

Sergio Borsani

A scuola con il Texas TI 99/4A



Il piacere del computer



Il piacere del computer

Il piacere del computer serie diretta da Mauro Boscarol

- 1 *Tom Rugg e Phil Feldman* 32 programmi con il PET
- 2 *Rich Didday* Intervista sul personal computer, hardware
- 3 *Tom Rugg e Phil Feldman* 32 programmi con l'Apple
- 4 *Ken Knecht* Microsoft Basic
- 5 *Paul M. Chirlian* Pascal
- 6 *Tom Rugg e Phil Feldman* 32 programmi con il TRS-80
- 7 *Rich Didday* Intervista sul personal computer, software
- 8 *Herbert D. Peckham* Imparate il Basic con il PET/CBM
- 9 *Karl Townsend e Merl Miller* Il personal computer come professione
- 10 *Karen Billings e David Moursund* Te ne intendi di computer?
- 11 *Thomas Dwyer e Margot Critchfield* Il Basic e il personal computer, uno: introduzione
- 12 *Don Inman e Kurt Inman* Imparate il linguaggio dell'Apple
- 13 *Thomas Dwyer e Margot Critchfield* Il Basic e il personal computer, due: applicazioni
- 14 *Luigi Pierro* Il manuale del CP/M
- 15 *Carlo Sintini* A scuola con il PET/CBM
- 16 *David Johnson-Davies* Il manuale dell'Atom
- 17 *David E. Schultz* Il libro del Commodore VIC 20
- 18 *Jim Huffman e Robert Bruce* Il "debug" nei personal computer
- 19 *John M. Nevison* Programmazione in Basic per l'uomo d'affari
- 20 *Mark Harrison* Imparate il Basic con lo ZX81
- 21 *Ronald W. Andersen* Dal Basic al Pascal
- 22 *Herbert D. Peckham* Imparate il Basic con il Texas TI 99/44
- 23 *Sergio Borsani* A scuola con il Texas TI 99/4A

Sergio Borsani

A scuola con il Texas TI 99/4A



franco muzzio & c. editore

Prima edizione: gennaio 1984
ISBN 88-7021-242-4

© 1984 franco muzzio & c. editore
Via Bonporti 36, 35141 Padova, tel. 049/661147-661873
Tutti i diritti sono riservati.

Indice generale

7 Premessa.

11 Matematica elementare.

I numeri primi e la scomposizione in fattori.

L'algoritmo euclideo.

Massimo Comun Divisore e Minimo Comune Multiplo.

Frazioni.

Interesse e sconto.

34 Geometria elementare.

Il numero π greco.

Il volume del cono.

Il teorema di Pitagora.

Angoli e distanze.

57 Rappresentazioni grafiche.

Caratteristiche grafiche del TI-99/4A.

Grafici a strisce.

Punti e rette sul piano cartesiano.

Geometria analitica: parabole.

87 Ancora matematica.

Studio di una funzione e sue derivate.

Soluzione delle equazioni algebriche.

Determinanti e soluzioni di sistemi lineari.
Integrali definiti.

117 Calcolo combinatorio.

Introduzione.
Disposizioni semplici e permutazioni.
Combinazioni semplici.
Verifica di una distribuzione Gaussiana.
La città degli esagoni.

148 Fisica.

Le forze.
Cinematica.
Moto ad accelerazione costante.
Orbite dei satelliti artificiali.
Conversione delle unità di misura.

186 Tra l'utile e il dilettevole.

Città d'Italia.
Archivio scacchistico.
Memory.
L'ultimo perde.

207 Indice analitico

Premessa

La sempre più vasta diffusione degli home computer induce a pensare che siano in molti a possedere una sufficiente conoscenza del Basic e quindi ormai in grado di scrivere i propri programmi. Inoltre è certo che un home computer non è solo un piacevole hobby o uno svago ma è uno strumento scientifico e, con del software adeguato, può consentire un approccio nuovo ed eccitante anche nei confronti di quelli che possono considerarsi classici argomenti di studio.

A buon diritto esso sta trovando un suo spazio anche nel mondo della scuola e si può prevedere che le nuove generazioni di insegnanti sentiranno l'esigenza di non essere soltanto gli utilizzatori passivi di questo nuovo strumento didattico.

Tuttavia l'uso del computer in campo matematico e scientifico non consente improvvisazioni ed è necessario che il lettore venga gradualmente portato a formarsi una logica di base specifica per tali applicazioni.

Un libro generico non sempre assolve a questo scopo; spesso costringe ad adattamenti e per forza di cose non consente di utilizzare in modo armonico tutte le risorse del computer.

Al contrario i programmi qui raccolti sono stati scritti espressamente per il TI 99/4A home computer della Texas Instruments Inc. e ne sfruttano appieno le capacità; inoltre, non essendo adattamenti da altre versioni di Basic, contengono alcune novità e soluzioni originali.

Per consentirne l'uso alla maggior parte dei lettori, quasi tutti i programmi sono stati scritti in TI Basic e solo alcuni in Extended Basic che,

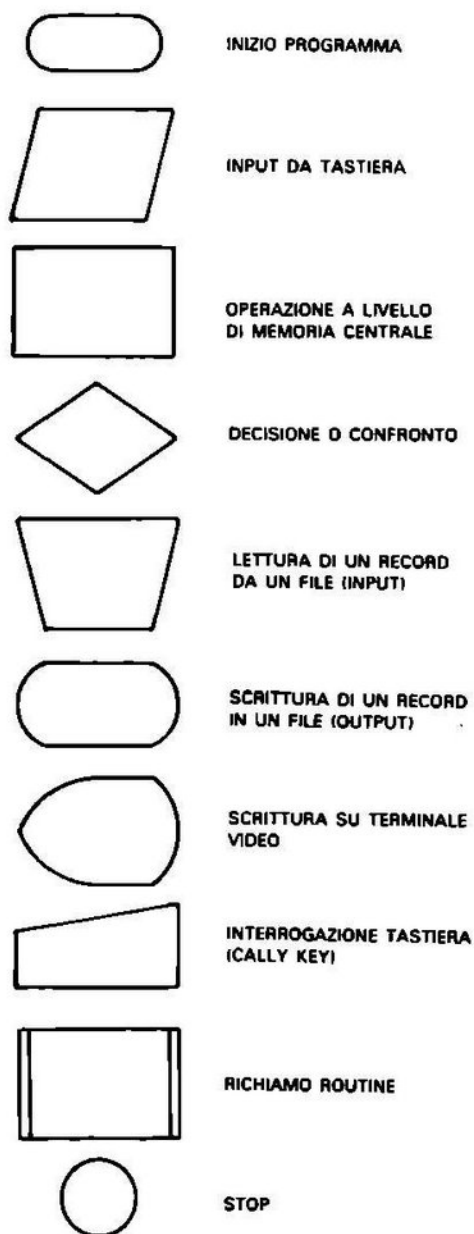


Fig. 1

come è noto, richiede il modulo di comando acquistabile separatamente. I temi sono presentati in ordine di difficoltà crescente, essendo i primi generalmente trattati negli studi medi inferiori mentre gli ultimi affrontano argomenti specifici degli studi superiori.

Costituisce un'eccezione l'ultimo capitolo che riguarda applicazioni varie e giochi.

Non sempre questo ordine logico rispecchia le difficoltà insite nella tecnica di programmazione; il primo programma, ad esempio, pur trattando l'argomento elementare della scomposizione in fattori primi, contiene degli algoritmi che non sono certamente di facile comprensione.

Ogni listato è preceduto da una trattazione generale dell'argomento da un punto di vista matematico, dall'esposizione delle tecniche utilizzate e dalle indicazioni per un corretto uso del programma; segue poi l'esame dettagliato delle istruzioni.

A volte, per meglio illustrare l'algoritmo utilizzato, è stato introdotto un diagramma a blocchi; spesso viene mostrato anche un esempio pratico di utilizzazione del programma e di ciò che appare sul video in risposta a particolari quesiti.

I programmi non sono eccessivamente lunghi, raramente occupano più di 10 Kbyte. Tuttavia dovrà fare molta attenzione chi intendesse ampliarli. Questo è importante soprattutto per chi usa il registratore a cassette e l'Expansion Sistem poiché con tale configurazione i programmi su nastro non possono superare i 12 Kbyte (vedi il manuale dell'*Extended Basic Memory Expansion Unit and Cassette-based Programs*).

Nella maggior parte dei casi i programmi sono protetti da eventuali errori dell'utente così da evitare che vengano introdotti errati valori alfanumerici che potrebbero provocare una interruzione. Tuttavia, quando si chiede una risposta nella forma S/N è quasi sempre necessario usare la lettera maiuscola; come si sa le lettere maiuscole e minuscole hanno due differenti numeri di codice ASCII, pertanto, durante l'esecuzione dei programmi, è buona norma tenere il tasto ALPHA LOCK in posizione abbassata.

Per chi non avesse familiarità con i diagrammi a blocchi, la figura 1 riporta i simboli adottati ed il loro significato.

Matematica elementare

I numeri primi e la scomposizione in fattori

Come è noto fin dagli studi secondari inferiori, un numero intero maggiore di uno si dice primo se non ammette divisori diversi da se stesso e da 1. Questo concetto apparentemente banale ha, in realtà, impegnato generazioni di matematici ed alcuni problemi ad esso legati sono tuttora irrisolti. Uno di questi va sotto il nome di "ipotesi di Goldbach". L'affermazione che ogni numero pari, maggiore di 2, sia esprimibile come somma di due numeri primi non è stata né smentita né dimostrata in modo generale. È stata invece contraddetta l'ipotesi di Fermat che tutti i numeri generati dalla formula

$$F(n) = 2^{2^n} + 1$$

siano primi. Ad esempio, $2^{2^4} + 1 = 65\,537$ è primo, ma $2^{2^5} + 1$ è uguale al prodotto $641 \times 6\,700\,417$ e quindi non è primo.

Ogni numero intero può essere rappresentato come prodotto di numeri primi. Ad esempio, $120 = 2^3 \times 3 \times 5$. Questa operazione viene denominata "scomposizione in fattori primi".

Ed ora vediamo il programma che consente la scomposizione di qualsiasi numero.

```

100 REM      SCOMPOSIZIONE
110 REM      IN FATTORI
120 REM      *****
130 REM      DI: SERGIO BORSANI
140 REM      VERSIONE: TI BASIC
150 REM
160 CALL CLEAR
170 CALL CHAR(144,"F")
180 CALL COLOR(15,13,13)
190 PRINT TAB(8);"SCOMPOSIZIONE":
200 PRINT TAB(9);"IN FATTORI.";:
210 CALL VCHAR(10,8,144,17)
220 CALL VCHAR(11,8,144,3)
230 CALL VCHAR(11,24,144,5)
240 CALL VCHAR(11,8,144,17)
250 CALL CHAR(131,"F0909090F0")
260 CALL CHAR(131,"2060202020")
270 CALL CHAR(132,"F010F080F0")
280 CALL CHAR(133,"F010F010F0")
290 CALL CHAR(134,"B08090F01")
300 CALL CHAR(135,"F080F010F")
310 CALL CHAR(136,"F080F090F")
320 CALL CHAR(137,"F010204080")
330 CALL CHAR(138,"F090F090F0")
340 CALL CHAR(139,"F090F010F0")
350 DATA 2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79
360 DATA 81,89,97,101,103,107,109,113,127,131,137,139,149,151,157,163
370 DATA 167,173,179,181,191,193,197,199,211,223,227,229,233,239,241,251
380 DATA 257,263,269,271,277,281,283,293,307,311,313,317,331,337,347,349
390 DATA 353,359,367,373,379,383,389,397,401,409,419,421,431,433,439,443
400 DATA 449,457,461,463,467,479,487,491,499,503,509,521,523,541
410 RESTORE 350
420 PRINT "SCRIVI IL NUMERO DA SCOMPOR-": "RE IN FATTORI.":
430 INPUT "N = ":N
440 PRINT
450 PRINT N;" = "
460 FOR I=1 TO 100
470 READ P
480 L=0
490 T=0
500 Q=N/P
510 IF Q=INT(Q) THEN 540
520 IF T=1 THEN 580
530 GOTO 630
540 T=1
550 L=L+1
560 N=Q
570 GOTO 500
580 PRINT STR$(P);
590 IF L=1 THEN 610
600 GOSUB 940
610 IF N=1 THEN 870
620 PRINT CHR$(42);
630 NEXT I
640 PRIMO=547
650 IF PRIMO>SQR(N) THEN 860
660 L=0
670 T=0
680 Q=N/PRIMO
690 IF Q=INT(Q) THEN 740
700 IF T=1 THEN 780
710 PRIMO=PRIMO+2
720 GOSUB 990
730 GOTO 660
740 T=1
750 L=L+1
760 N=Q
770 GOTO 680
780 PRINT STR$(PRIMO);
790 IF L=1 THEN 810
800 GOSUB 940
810 IF N=1 THEN 870
820 PRINT CHR$(42);
830 PRIMO=PRIMO+2
840 GOSUB 990
850 GOTO 650
860 PRINT STR$(N)
870 PRINT :
880 PRINT "VUOI CONTINUARE? (S/N)";
890 CALL KEY(0,KK,SS)
900 IF SS=0 THEN 890
910 IF (KK<>78)*(KK<>83) THEN 890
920 CALL CLEAR
930 IF KK=83 THEN 410 ELSE 1130
940 FOR J=1 TO LEN(STR$(L))
950 M=VAL(SEG$(STR$(L),J,1))

```

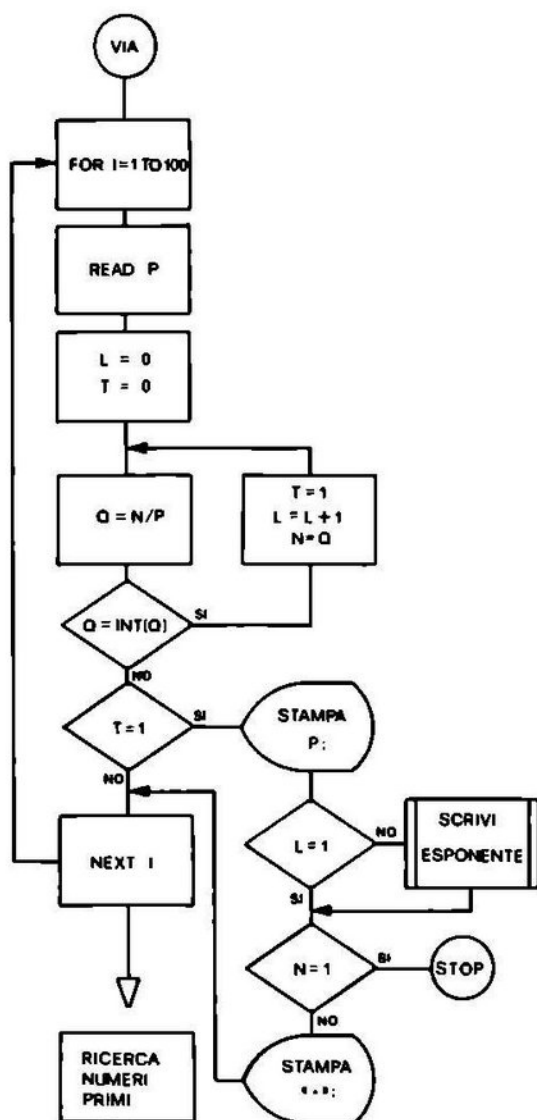



Fig. 2

```

960 PRINT CHR$(130+M)
970 NEXT J
980 RETURN
990 RESTORE 350
1000 FOR I=2 TO 100
1010 READ P
1020 IF P>SQR(PRIMO) THEN 1140
1030 IF PRIMO/P<>INT(PRIMO/P) THEN 1060
1040 PRIMO=PRIMO+2
1050 GOTO 990
1060 NEXT I
1070 Z=547
1080 IF Z>SQR(PRIMO) THEN 1140
1090 IF PRIMO/Z<>INT(PRIMO/Z) THEN 1120
1100 PRIMO=PRIMO+2
1110 GOTO 990
1120 Z=Z+2
1130 GOTO 1080
1140 RETURN
1150 END

```

Un programma per la scomposizione in fattori deve contenere una routine per la ricerca dei numeri primi, infatti i divisori di un numero dato vanno cercati solo nel loro insieme. Teoricamente per stabilire se un numero è primo basta verificare che non abbia altri divisori che se stesso. La verifica, se condotta in modo estensivo, comporta tempi di elaborazione piuttosto lunghi e si rende necessaria l'applicazione di alcuni concetti volti ad ottimizzare il calcolo.

Primo, ad eccezione del 2 tutti i numeri primi vanno ricercati tra i numeri dispari. Secondo, la ricerca di eventuali divisori di un numero non è necessario che si spinga oltre la radice quadrata del numero stesso. Infatti, immaginiamo ad esempio di voler stabilire se 47 è primo; verificato che non è divisibile per due, per 3 e per 5, è inutile controllare la divisibilità per 7, poiché se 7 fosse un divisore, allora 47 dovrebbe essere scomponibile nella forma 7×5 o $7 \times 3 \times 2$, dove compaiono anche fattori che in precedenza avevamo già scartato; quindi 47 è senz'altro primo.

Il programma inizia con la definizione dei caratteri che rappresentano gli esponenti (linee 250-340); successivamente, per raggiungere una maggiore efficienza, sono stati memorizzati con istruzioni DATA i primi cento numeri primi fino a 541.

Possiamo distinguere due sezioni principali, molto simili tra di loro: la prima utilizza i numeri primi già memorizzati nel programma (vedi figura 1), la seconda effettua la ricerca dei numeri primi superiori a 541.

Consideriamo la routine denominata STAMPA. N rappresenta inizialmente il numero da scomporre in fattori, successivamente è uguale alla parte che rimane da scomporre. P contiene i numeri primi in ordine crescente (2, 3, 5, 7, ecc.); L è un contatore che contiene l'esponente con il quale compare un fattore primo. Se si trova un numero primo divisore di N, L viene incrementato di una unità ed N prende il valore del quoziente. Finché la divisione è esatta si continua a dividere per lo stesso numero primo, altrimenti si passa al numero primo successivo. Si procede in questo modo finché non sia $N=1$.

Quando la variabile ha raggiunto il valore di 541, viene posto

PRIMO=547 e successivamente si incrementa di 2, poiché così si rimane nell'ambito dei numeri dispari. La routine alle linee 1000-1150 ha lo scopo di verificare se il nuovo valore è effettivamente un numero primo oppure no. Per fare questo il valore della variabile PRIMO viene diviso successivamente per i numeri primi memorizzati con le istruzioni DATA e, se non si trovano divisori tra questi, la ricerca si estende a tutti i numeri dispari superiori a 541. Solo se non si trovano divisori, PRIMO viene effettivamente considerato un numero primo e passa nella routine STAMPA per verificare se è un divisore del residuo N.

Con gli accorgimenti adottati e se i fattori primi non sono molto alti, il tempo di esecuzione si riduce generalmente a pochi secondi. La memoria richiesta è di 2,7 Kbyte.

Esempio pratico:

RUN

SCRIVI IL NUMERO DA SCOMPOR-
RE IN FATTORI.

N = 1440

1440 = 2⁵*3²*5

VUOI CONTINUARE? (S/N)

Tempo di esecuzione: 1 secondo e 1/2.

L'algoritmo euclideo

Supponiamo di eseguire un'ordinaria divisione tra i numeri interi 628 e 24; si ha il quoziente 26 ed il resto 4. È soddisfatta la seguente uguaglianza

$$628 = 26 \times 24 + 4$$

Si può dimostrare che i divisori comuni di 628 e 24 sono anche divisori comuni di 24 e 4. Ripetendo il procedimento con questi due numeri si ottiene il quoziente 6 ed il resto 0, pertanto

$$24 = 6 \times 4 + 0$$

Anche in questo caso i divisori comuni di 24 e 4 sono divisori di 4 e 0.

Si deduce che 4 è il massimo comun divisore di 628 e 24.
 Riportiamo il diagramma a blocchi relativo a questo algoritmo.

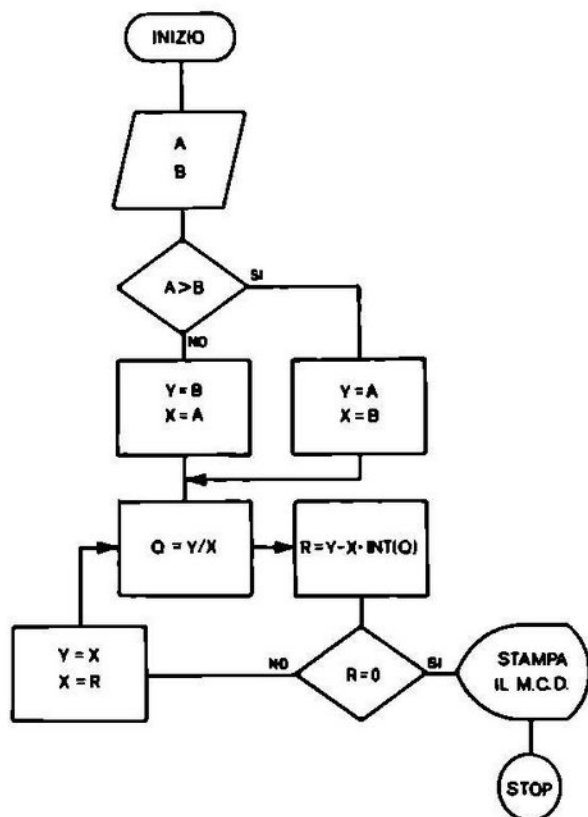


Fig. 3

Segue il programma in Basic che potrà essere utilizzato come routine per trovare il mcd tra due numeri e per ridurre ai minimi termini una frazione.

```

100 CALL CLEAR
110 PRINT TAB(6); "ALGORITMO EUCLIDEO": "::::::::::
120 FOR T=1 TO 500
130 NEXT T
140 PRINT "E' UN METODO PER TROVARE IL": "M.C.D. TRA DUE NUMERI": "
150 PRINT "NEL CORSO DEL PROGRAMMA PRE-MI UN TASTO PER PROSEGUIRE.": "
160 PRINT "ORA SCRIVI I DUE NUMERI": "
170 INPUT "A = ": A
180 INPUT "B = ": B
190 PRINT
200 PRINT "DIVIDO IL MAGGIORE PER IL": "MINORE": "
210 IF A > B THEN 250

```

```

220 Y=B
230 Y=A
240 GOTO 270
250 Y=A
260 T=B
270 Q=Y/X
280 R=Y-Y*INT(Q)
290 PRINT Y:"X:";X:"=";INT(Q)
300 PRINT "CON IL PESTO DI":R:
310 CALL KEY(0,1,5)
320 IF S=0 THEN 310
330 IF R=0 THEN 380
340 PRINT "PROSEGUO DIVIDENDO IL PRECE-DENTE DIVISORE PER IL RESTO":
350 Y=X
360 X=R
370 GOTO 270
380 PRINT "POICHE' IL RESTO E' UGUALE 0"
390 PRINT "ZERO,";X;"E' IL M.C.D.":
400 PRINT "M.C.D. (";STR$(A);";";STR$(B);")=";X:
410 PRINT "VUOI CONTINUARE" (5/N):
420 CALL KEY(0,1,5)
430 IF S=0 THEN 420
440 IF I=8 THEN 100
450 IF I=78 THEN 470
460 GOTO 420
470 CALL CLEAR
480 END

```

Dopo la presentazione e le istruzioni, vengono accettati i due numeri dei quali si vuol trovare il mcd (170-180). Le linee 190-260 hanno lo scopo di attribuire alla Y il maggiore dei due numeri. La variabile Q è il quoziente mentre R è il resto. La linea 330 manda alle conclusioni se il resto è zero, altrimenti il precedente divisore diventa il nuovo dividendo e il resto diventa il nuovo divisore (350-360). Le conclusioni sono stampate alle righe 380-400.

Massimo comune divisore e minimo comune multiplo

L'algoritmo euclideo fornisce un metodo rapido e sicuro per trovare il mcd tra due numeri, tuttavia, quando i numeri sono più di due, bisogna seguire una via diversa.

Poiché il mcd di più numeri deve essere minore o al massimo uguale al più piccolo di essi, si verifica se questo è divisore di tutti gli altri numeri; se lo è, esso è il mcd, se non lo è, si divide successivamente per 2, 3, 4 ecc. finché il risultato è un numero intero, dopo di che si verifica nuovamente se questo divisore è anche divisore di tutti gli altri numeri. Per il mcm le considerazioni sono analoghe. Infatti il mcm è o il maggiore dei numeri dati o un suo multiplo.

In Basic per conoscere se un numero è multiplo di un altro numero, si verifica se è soddisfatta la seguente condizione:

$$M/N = \text{INT}(M/N)$$

dove M e N sono i numeri dati. La funzione INT restituisce la parte intera del quoziente.

```

100 REM MASSIMO COMUNE DIVISORE
110 REM E
120 REM MINIMO COMUNE MULTIPLO.
130 REM *****
140 REM DI: SERGIO BORSANI
150 REM VERSIONE: TI BASIC
160 REM
170 CALL CLEAR
180 CALL CHAR(128,"F")
190 CALL COLOR(13,11,11)
200 PRINT TAB(4);"MASSIMO COMUN DIVISORE":
210 PRINT TAB(14);"E":
220 PRINT TAB(4);"MINIMO COMUNE MULTIPLO":
230 CALL HCHAR(10,4,128,26)
240 CALL VCHAR(11,4,128,7)
250 CALL VCHAR(11,24,128,7)
260 CALL HCHAR(18,4,128,26)
270 FOR T=1 TO 500
280 NEXT T
290 PRINT "PREMI:":
300 PRINT "1. PER IL M.C.D.":
310 PRINT "2. PER IL m.c.m.":
320 PRINT "3. PER FINIRE"
330 CALL KEY(0,K,S)
340 IF S=0 THEN 330
350 IF K=49 THEN 390
360 IF K=50 THEN 710
370 IF K=51 THEN 1130
380 GOTO 330
390 CALL CLEAR
400 PRINT "M.C.D.":
410 CALL HCHAR(24,3,128,28)
420 PRINT ":
430 PRINT "DI QUANTI NUMERI VUOI TROVARE IL M.C.D. "
440 INPUT N
450 PRINT
460 IF N<10 THEN 490
470 PRINT "SPIACENTE, NON PIU' DI 10.":
480 GOTO 430
490 PRINT "SCRIVI UN NUMERO PER VOLTA, POI PREMI ENTER":
500 INPUT X(1)
510 MINI=X(1)
520 FOR I=2 TO N
530 INPUT X(I)
540 IF MINI<X(I) THEN 560
550 MINI=X(I)
560 NEXT I
570 K=0
580 K=K+1
590 Q=MINI/K
600 IF Q<INT(Q) THEN 580
610 FOR I=1 TO N
620 IF X(I)/Q=INT(X(I)/Q) THEN 640
630 GOTO 580
640 NEXT I
650 PRINT
660 PRINT "M.C.D. =";Q:
670 GOSUB 1080
680 IF K=83 THEN 390
690 CALL CLEAR
700 GOTO 290
710 CALL CLEAR
720 PRINT "m.c.m.":
730 CALL HCHAR(24,3,128,28)
740 PRINT ":
750 PRINT "DI QUANTI NUMERI VUOI TROVARE IL m.c.m. ";
760 INPUT N
770 PRINT
780 IF N<10 THEN 810
790 PRINT "SPIACENTE, NON PIU' DI 10.":
800 GOTO 750
810 PRINT "SCRIVI UN NUMERO PER VOLTA, POI PREMI ENTER":
820 INPUT X(1)
830 MAX=X(1)
840 FOR I=2 TO N
850 INPUT X(I)
860 IF MAX<X(I) THEN 880
870 MAX=X(I)
880 NEXT I
890 K=0
900 K=K+1
910 P=MAX*K
920 FOR I=1 TO N
930 IF P/X(I)=INT(P/X(I)) THEN 950
940 GOTO 900

```

```

950 NEXT I
960 PRINT
970 PRINT "a.c.m. =":P::
980 PRINT
990 PRINT P;"="
1000 FOR I=1 TO N
1010 PRINT TAB(LEN(STR$(P))+5);X(I);"=";P/X(I)
1020 NEXT I
1030 PRINT "::
1040 GOSUB 1080
1050 IF K=83 THEN 710
1060 CALL CLEAR
1070 GOTO 290
1080 PRINT "VUOI CONTINUARE? (S/N) "
1090 CALL KEY(0,K,S)
1100 IF S=0 THEN 1090
1110 IF (K<>78) & (K<>83) THEN 1090
1120 RETURN
1130 CALL CLEAR
1140 END

```

Alle linee 290-320 viene chiesto se si vuole ottenere il mcd o il mcm. La scelta è accettata con un CALL KEY, pertanto non è necessario premere il tasto ENTER. Nella linea 440 bisogna specificare di quanti numeri si vuole ottenere il mcd. N è stato posto inferiore a 10. Successivamente si scrivono i numeri uno per volta e questi vengono raccolti nella variabile con indice X(I); nella variabile MINI si carica il minore dei numeri. K è una variabile che assume i valori 1, 2, 3, 4..., Q è il quoziente della divisione MINI/K. Alla linea 620 si controlla se Q è divisore di tutti gli altri numeri dati e quando si verifica tale circostanza Q è il massimo comune divisore.

Alla linea 710 inizia la sezione per il calcolo del mcm. Il procedimento è simile al precedente. Alle linee 820-880 si chiede l'input dei numeri, il massimo di questi viene memorizzato nella variabile MAX. K assume successivamente i valori 1, 2, 3, 4..., P è il prodotto $MAX \cdot K$. Quando P è multiplo di tutti i numeri dati, allora è il minimo comune multiplo. Il suo valore viene stampato alla linea 270.

Poiché talvolta può rivelarsi utile, alle linee 1000-1020 viene stampato anche quante volte ogni numero è contenuto nel mcm.

Frazioni

Il programma svolge le più comuni operazioni con i numeri frazionari, nell'ordine: trasformazione di una frazione in un numero decimale, trasformazione di un numero decimale in frazione, sia esso limitato o illimitato periodico, semplificazione, ricerca di una frazione equivalente ad una data, confronto e le quattro operazioni aritmetiche.

L'argomento generalmente viene affrontato al termine del ciclo di istruzione primaria e nel primo anno della scuola media, pertanto non è necessario esporre le regole di calcolo che sono già note a tutti. Anche la programmazione non presenta particolari difficoltà. Tuttavia è necessario un certo impegno nel prevedere errori da parte dell'utente che po-


```

490 FOR I=2 TO K+1
500 PRINT TAB(I);CHR$(128);
510 NEXT I
520 PRINT TAB(K+3);"=";X
530 PRINT B::
540 GOSUB 3580
550 IF KK=83 THEN 410 ELSE 250
560 CALL CLEAR
570 PRINT "TRASFORMARE UN NUMERO":"DECIMALE IN FRAZIONE":
580 CALL MCHAR(24,3,136,28)
590 PRINT ::
600 PRINT "IL NUMERO DECIMALE E' PERIO-":"DICO? (S/N)":
610 CALL KEYIO, KK, SS)
620 IF SS=0 THEN 610
630 IF KK=78 THEN 640
640 IF KK=83 THEN 870
650 GOTO 610
660 PRINT "SCRIVI IL NUMERO DECIMALE":
670 INPUT "N = ":N
680 LL=LEN(STR$(N))
690 D=LEN(STR$(N-INT(N)))-1
700 A=N*10^D
710 B=10^D
720 GOSUB 3630
730 PRINT ::
740 PRINT TAB(4+LL);NUME
750 PRINT N;"=";
760 IF NUME>DEND THEN 790
770 K=LEN(STR$(DEND))
780 GOTO 800
790 K=LEN(STR$(NUME))
800 FOR I=5+LL TO 4+LL+K
810 PRINT TAB(I);CHR$(128);
820 NEXT I
830 PRINT
840 PRINT TAB(4+LL);DEND::
850 GOSUB 3580
860 IF KK=83 THEN 560 ELSE 250
870 PRINT ::
880 PRINT "SCRIVI IL NUMERO SENZA RIPE-":"TERE IL PERIODO":
890 INPUT "N = ":X
900 PRINT
910 PRINT "QUANTE CIFRE FORMANO IL"
920 INPUT "PERIODO? ":PER
930 PRINT ::
940 ANT=LEN(STR$(X-INT(X)))-PER-1
950 LL=LEN(STR$(X))
960 DEC=ANT+PER
970 N=X*10^DEC-INT(X*10^ANT)
980 D=9
990 IF PER<2 THEN 1030
1000 FOR I=1 TO PER-1
1010 D=D*10+D
1020 NEXT I
1030 D=D*10^ANT
1040 FOR I=1 TO PER
1050 PRINT TAB(1+LL+I-PER);CHR$(129);
1060 NEXT I
1070 PRINT TAB(LL+4);N
1080 PRINT X;"=";
1090 IF X>1 THEN 1120
1100 L=LEN(STR$(N))
1110 GOTO 1130
1120 K=LEN(STR$(D))
1130 FOR I=0 TO K
1140 PRINT TAB(LL+I+3);CHR$(128);
1150 NEXT I
1160 PRINT
1170 PRINT TAB(LL+4);D::
1180 A=N
1190 B=D
1200 FF=1
1210 GOTO 1290
1220 FF=0
1230 IF KK=83 THEN 560 ELSE 250
1240 CALL CLEAR
1250 PRINT "SEMPLIFICARE UNA FRAZIONE":
1260 CALL MCHAR(24,3,136,28)
1270 PRINT ::
1280 GOSUB 3780
1290 GOSUB 3630
1300 PRINT A;
1310 GOSUB 3840
1320 PRINT TAB(4+K);NUME
1330 FOR I=2 TO K+1
1340 PRINT TAB(I);CHR$(128);

```

```

1350 NEXT I
1360 PRINT " = ";
1370 IF NUME>DENO THEN 1400
1380 K=LEN(STR$(DENO))
1390 GOTO 1410
1400 K=LEN(STR$(NUME))
1410 FOR J=I+3 TO I+K+2
1420 PRINT TAB(J);CHR$(128);
1430 NEXT J
1440 PRINT
1450 PRINT B;TAB(I+2);DENO::
1460 GOSUB 3580
1470 IF FF=1 THEN 1220
1480 IF KK=83 THEN 1240 ELSE 250
1490 CALL CLEAR
1500 PRINT "FRAZIONI EQUIVALENTI":
1510 CALL HCHAR(24,3,136,28)
1520 PRINT ::
1530 PRINT "SCRIVI LA FRAZIONE": "CONOSCIUTA":
1540 GOSUB 3780
1550 PRINT TAB(13-LEN(STR$(A)));STR$(A);TAB(16);"M"
1560 GOSUB 3860
1570 FOR I=13-K TO 12
1580 PRINT TAB(I);CHR$(128);
1590 NEXT I
1600 PRINT TAB(14);" = ";CHR$(128)
1610 PRINT TAB(13-LEN(STR$(B)));STR$(B);TAB(16);"N":
1620 INPUT "COSA VUOI TROVARE? (M/N) ":R$
1630 PRINT
1640 IF R$="M" THEN 1680
1650 IF R$="N" THEN 1750
1660 PRINT "ERRORE! RIPROVA":
1670 GOTO 1620
1680 INPUT "N = ":N
1690 PRINT
1700 IF N<>0 THEN 1730
1710 PRINT "ERRORE! RIPROVA":
1720 GOTO 1620
1730 M=A*N/B
1740 GOTO 1780
1750 INPUT "M = ":M
1760 PRINT
1770 N=M*B/A
1780 GOSUB 3860
1790 K1=K
1800 IF M>N THEN 1830
1810 K2=LEN(STR$(N))
1820 GOTO 1840
1830 K2=LEN(STR$(M))
1840 PRINT TAB(13-LEN(STR$(A)));STR$(A);TAB(16+K2-LEN(STR$(M)));STR$(M)
1850 FOR I=13-K1 TO 12
1860 PRINT TAB(I);CHR$(128);
1870 NEXT I
1880 PRINT " = ";
1890 FOR I=16 TO 15+K2
1900 PRINT TAB(I);CHR$(128);
1910 NEXT I
1920 PRINT
1930 PRINT TAB(13-LEN(STR$(B)));STR$(B);TAB(16+K2-LEN(STR$(N)));STR$(N)::
1940 GOSUB 3580
1950 IF KK=83 THEN 1490 ELSE 250
1960 CALL CLEAR
1970 PRINT "CONFRONTO":
1980 CALL HCHAR(24,3,136,28)
1990 PRINT ::
2000 PRINT "QUANTE FRAZIONI DEVI"
2010 INPUT "CONFRONTARE? ":N
2020 PRINT
2030 IF N<=10 THEN 2060
2040 PRINT "SPIACENTE, NON PIU' DI 10":
2050 GOTO 2000
2060 FOR I=1 TO N
2070 PRINT "FRAZIONE";I;": "
2080 GOSUB 3780
2090 NN(I)=A
2100 DD(I)=B
2110 FRZ(I)=A/B
2120 VD(I)=FRZ(I)
2130 PRINT
2140 NEXT I
2150 SW=0
2160 FOR I=1 TO N-1
2170 IF VD(I)<=VD(I+1) THEN 2220
2180 FF=VD(I)
2190 VD(I)=VD(I+1)
2200 VD(I+1)=FF

```

```

2210 SW=1
2220 NEXT I
2230 IF SW=1 THEN 2150
2240 PRINT "LE FRAZIONI IN ORDINE": "CRESCENTE SONO":
2250 FOR I=1 TO N
2260 FOR J=1 TO N
2270 IF VD(I)=FRZ(J) THEN 2290
2280 NEXT J
2290 PRINT STR$(I); " ";
2300 IF NN(J)<0 THEN 2320
2310 PRINT " ";
2320 PRINT STR$(NN(J)); "/" ; STR$(DD(J))
2330 NEXT I
2340 PRINT ::
2350 GOSUB 3380
2360 IF KK=83 THEN 1960 ELSE 250
2370 CALL CLEAR
2380 PRINT "ADDIZIONE E SOTTRAZIONE":
2390 CALL HCHAR(24,2,136,28)
2400 PRINT ::
2410 PRINT "QUANTE SONO LE FRAZIONI DA"
2420 INPUT "SCHIMARE? ":N
2430 PRINT
2440 IF N<10 THEN 2470
2450 PRINT "SPIACENTE, NON PIU' DI 10":
2460 GOTO 2410
2470 PRINT "PER L'ADDIZIONE FAI PRECEDE-": "RE IL NUMERATORE DAL SEGNO +"
2480 PRINT "PER LA SOTTRAZIONE FAI PRE-"
2490 PRINT "CEDERE IL NUMERATORE DAL SE-": "GNO -."::
2500 TN=0
2510 TD=1
2520 FOR I=1 TO N
2530 PRINT "FRAZIONE"; I; ":"
2540 GOSUB 3780
2550 NN(I)=A
2560 DD(I)=B
2570 TD=TD*DD(I)
2580 NEXT I
2590 PRINT ::
2600 FOR I=1 TO N
2610 C=TD/DD(I)
2620 TN=TN+NN(I)*C
2630 NEXT I
2640 IF NN(1)<0 THEN 2660
2650 PRINT " ";
2660 PRINT " "; STR$(NN(1)); "/" ; STR$(DD(1))
2670 IF N<2 THEN 2750
2680 FOR I=2 TO N
2690 IF NN(I)<0 THEN 2720
2700 PRINT " + ";
2710 GOTO 2730
2720 PRINT " - ";
2730 PRINT STR$(NN(I)); "/" ; STR$(DD(I))
2740 NEXT I
2750 CALL HCHAR(24,3,128,12)
2760 PRINT
2770 IF TN<0 THEN 2800
2780 PRINT " = 0":
2790 GOTO 2870
2800 A=TN
2810 B=TD
2820 GOSUB 3630
2830 IF DEND>0 THEN 2860
2840 DEND=-DEND
2850 NUME=-NUME
2860 PRINT " = "; STR$(NUME); "/" ; STR$(DEND)::
2870 GOSUB 3380
2880 IF KK=83 THEN 2370 ELSE 250
2890 CALL CLEAR
2900 PRINT "MOLTIPLICAZIONE":
2910 CALL HCHAR(24,3,136,28)
2920 PRINT ::
2930 PRINT "QUANTE FRAZIONI DEVI"
2940 INPUT "MOLTIPLICARE? ":N
2950 PRINT
2960 IF N<10 THEN 2990
2970 PRINT "SPIACENTE, NON PIU' DI 10":
2980 GOTO 2930
2990 TN=1
3000 TD=1
3010 FOR I=1 TO N
3020 PRINT "FRAZIONE"; I; ":"
3030 GOSUB 3780
3040 NN(I)=A
3050 DD(I)=B
3060 TN=TN*NN(I)

```

```

3070 TD=TD+DD(I)
3080 NEXT I
3090 PRINT " (";STR$(NN(1));"/";STR$(DD(1));")="
3100 IF N<2 THEN 3140
3110 FOR I=2 TO N
3120 PRINT " * (";STR$(NN(I));"/";STR$(DD(I));")="
3130 NEXT I
3140 CALL HCHAR(24,3,128,12)
3150 PRINT
3160 IF TN<>0 THEN 3190
3170 PRINT "= 0":GOTO 3180
3180 GOTO 3260
3190 A=TN
3200 B=TD
3210 GOSUB 3630
3220 IF DENO>0 THEN 3230
3230 NUME=-NUME
3240 DENO=-DENO
3250 PRINT " = ";STR$(NUME);"/";STR$(DENO):=
3260 GOSUB 3580
3270 IF KK=83 THEN 2890 ELSE 250
3280 CALL CLEAR
3290 PRINT "DIVISIONE":=
3300 CALL HCHAR(24,3,136,28)
3310 PRINT " "
3320 PRINT "VIENE ESEGUITA LA DIVISIONE":"TRA DUE FRAZIONI":=
3330 FOR I=1 TO 2
3340 PRINT "FRAZIONE"II":="
3350 GOSUB 3780
3360 NN(I)=A
3370 DD(I)=B
3380 NEXT I
3390 PRINT " "
3400 A=NN(1)*DD(2)
3410 B=DD(1)*NN(2)
3420 IF (A=0)+(B=0) THEN 3470
3430 GOSUB 3630
3440 IF DENO>0 THEN 3470
3450 NUME=-NUME
3460 DENO=-DENO
3470 PRINT " (";STR$(NN(1));"/";STR$(DD(1));
3480 PRINT " ) * (";STR$(NN(2));"/";STR$(DD(2));
3490 IF B<>0 THEN 3520
3500 PRINT " ) = IMPOSSIBILE":=
3510 GOTO 3560
3520 IF A<>0 THEN 3530
3530 PRINT " ) = 0":GOTO 3560
3540 GOTO 3560
3550 PRINT " ) = ";STR$(NUME);"/";STR$(DENO):=
3560 GOSUB 3580
3570 IF KK=83 THEN 3280 ELSE 250
3580 PRINT "VUOI CONTINUARE? (S/N)"
3590 CALL KEY(0, KK, SB)
3600 IF SB=0 THEN 3590
3610 IF (KK<>83)*(KK<>78) THEN 3590
3620 RETURN
3630 IF A>B THEN 3670
3640 Y=B
3650 X=A
3660 GOTO 3690
3670 Y=A
3680 X=B
3690 Q=Y/X
3700 R=Y-X*INT(Q)
3710 IF R=0 THEN 3730
3720 Y=X
3730 X=R
3740 GOTO 3690
3750 NUME=A/X
3760 DENO=B/X
3770 RETURN
3780 INPUT " NUMERATORE = ":A
3790 INPUT " DENOMINATORE = ":B
3800 PRINT
3810 IF B<>0 THEN 3850
3820 PRINT "ERRORE: IL DENOMINATORE NON"
3830 PRINT "PUO' ESSERE UGUALE A ZERO.":=
3840 GOTO 3790
3850 RETURN
3860 IF A>B THEN 3890
3870 K=LEN(STR$(B))
3880 GOTO 3900
3890 K=LEN(STR$(A))
3900 RETURN
3910 CALL CLEAR
3920 END

```

All'inizio del listato sono definiti alcuni caratteri grafici; il numero 128 è la linea di frazione. Il titolo occupa le linee seguenti (180-220). Il ciclo FOR-NEXT della variabile T serve semplicemente a mantenere il titolo sullo schermo per un certo tempo. Il menù (250-350) contiene 9 opzioni per accedere alle varie sezioni del programma.

Nella linea 380 si controlla che il tasto premuto sia compreso tra 1 e 9, i quali hanno come numero di codice ASCII 49 e 57; la scrittura, che a taluni potrebbe sembrare strana, simula la funzione OR che non è disponibile in TI Basic. Ricordo che il computer attribuisce il valore -1 ad un'uguaglianza o disuguaglianza vera ed il valore 0 ad una uguaglianza o disuguaglianza falsa, inoltre l'invio al numero di linea specificato dopo la parola THEN avviene solo se il valore trovato è -1. La prima sezione è contenuta nelle linee 410-550; in essa si calcola il valore decimale corrispondente ad un numero frazionario. Per l'input dei dati si va alla routine 3780 (GOSUB di linea 450); alla linea 460 viene calcolato il valore decimale (variabile X). Il risultato è preceduto dalla frazione scritta in modo convenzionale, con il numeratore, la linea di frazione ed il denominatore uno sull'altro e non sulla stessa riga. La routine 3860 (GOSUB di linea 480) serve a determinare la lunghezza della linea di frazione. Nella seconda sezione (560-1230) si trasforma un numero decimale in frazione. Si fa distinzione se il numero è decimale limitato o periodico. Le regole di calcolo sono quelle note; ad esempio, un numero periodico corrisponde ad una frazione che ha per numeratore il numero, senza la virgola, meno la parte non periodica e per denominatore un numero costituito da tanti nove quante sono le cifre del periodo seguiti da tanti zeri quante sono le cifre dell'antiperiodo. X (linea 890) è il numero periodico, PER (linea 920) è il numero delle cifre che formano il periodo, ANT (linea 940) è il numero delle cifre che formano l'antiperiodo, LL è la lunghezza dell'intero numero, DEC la lunghezza della parte decimale, N è il numeratore calcolato sottraendo la parte non periodica all'intero numero senza la virgola, infine D è il denominatore, calcolato alle linee 980-1030.

La stampa occupa le linee 1040-1230; dopo che è stata scritta una prima frazione, viene posto a 1 il flag FF e si passa alla sezione successiva dove si ristampa la stessa frazione ridotta ai minimi termini. Subito dopo il flag viene azzerato. Alle linee 1240-1480 si trova la sezione per semplificare una frazione; il calcolo, svolto nella routine 3630 (GOSUB di linea 1290), segue l'algoritmo euclideo. La sezione per la ricerca delle frazioni equivalenti occupa le linee 1490-1950. L'input dei dati è svolto ancora dalla routine 3780. Lo sforzo maggiore consiste nello scrivere le due frazioni equivalenti nel modo convenzionale, una a fianco dell'altra, separate dal segno "=".

La lunghezza delle linee di frazione ed il giusto incolonnamento sono fatti in base alla lunghezza dei quattro termini (linee 1780-1830). La li-

nea 1840 stampa i due numeratori, le successive le due linee di frazione e la riga 1930 stampa i due numeratori.

Dalla linea 1960 alla 2360 si effettua il confronto tra frazioni, cioè viene fatto il loro ordinamento dalla minore alla maggiore in base ai corrispondenti valori decimali.

Le sezioni seguenti sono dedicate alle quattro operazioni: addizione e sottrazione da linea 2370 a 2880, la moltiplicazione da 2890 a 3270 e la divisione da 3280 a 3570. Le principali variabili utilizzate in quest'ultima parte sono: NN(I) e DD(I), numeratori e denominatori delle frazioni in input; TD, denominatore comune, prodotto di tutti i denominatori; C, quante volte ogni numeratore è contenuto nel denominatore comune (linea 2610); TN, somma dei numeratori.

In fondo al programma sono state poste le routine. In 3580 quella per porre la domanda se si vuole continuare; la moltiplicazione alla linea 3610 simula l'istruzione IF...AND che non è disponibile in TI Basic. L'algoritmo di Euclide occupa le linee 3630-3770; in entrata abbiamo le variabili A e B, in uscita NUME e DENO. La routine 3780, come è già stato detto, serve per l'input di una frazione; in essa si controlla che il denominatore sia diverso da zero. L'ultima, righe 3860-3900, serve per determinare la lunghezza K di una linea di frazione.

Il programma occupa circa 7,6 Kbyte di memoria.

Interesse e sconto

Qual era il prezzo di listino se, dopo aver ottenuto uno sconto dell'8%, si è pagata una somma di lire 23 000? Quale aliquota percentuale è stata applicata alla somma di lire 85 000 per ottenere un importo di lire 97 000? Queste ed altre simili domande trovano una risposta nelle prime due sezioni del programma, dedicate, per l'appunto, allo sconto e all'interesse semplice. Il calcolo generalmente viene affrontato nella seconda classe della scuola media e consiste nel risolvere una proporzione. Ad esempio, considerando il primo problema, possiamo scrivere:

$$x : 23\,000 = 100 : 92$$

da cui si ricava:

$$x = \frac{23\,000 \times 100}{92} = 25\,000$$

L'interesse derivante da un deposito bancario, se non viene ritirato, viene capitalizzato, cioè sommato al capitale iniziale. Se, ad esempio, il capitale è di lire 1 000 000 ed il tasso di interesse è del 14%, dopo un anno

l'interesse ammonta a lire 140 000. L'anno successivo il tasso del 14% non è più applicato al milione iniziale ma a lire 1 140 000 che costituiscono quello che viene chiamato montante. Al passare degli anni il capitale aumenta secondo una legge esponenziale.

Ecco la formula per calcolare l'interesse composto:

$$M = C(1 + R/V)^{(V \times T)}$$

M è il montante, C il capitale iniziale, R è il tasso d'interesse, V è il numero di volte che, in un anno, l'interesse viene composto al capitale, T è il tempo, espresso in anni. Nel programma tale funzione è definita alla linea 370. Ricordo che in TI Basic il nome della funzione può essere seguito da una variabile posta tra parentesi. Altri tipi di Basic ammettono più variabili tra parentesi.

La sezione è arricchita dalla rappresentazione dei valori, anno per anno, in forma grafica e per tabulazione. Il grafico, un diagramma a strisce, è particolarmente efficace per comprendere il significato di una legge di accrescimento esponenziale. Una particolare routine che verrà spiegata nel terzo capitolo, permette di far variare, pixel per pixel, la lunghezza delle strisce che formano il grafico, le quali altrimenti varierebbero carattere per carattere (un carattere è lungo 8 pixel), rendendo più approssimativa la rappresentazione.

Per dare una dimensione reale al tema trattato è stata aggiunta una parte che consente di calcolare capitali, riferiti ad anni diversi, aventi lo stesso potere d'acquisto. Ad esempio, la somma di lire 800 000 del 1975, a causa dell'inflazione, equivale a più di 2 700 000 del 1983. I coefficienti necessari per il calcolo sono quelli medi annui desunti dalla rivista *Capital*. Essi coprono l'intervallo di tempo che va dal 1970 al 1983 e sono riferiti al 1982; quello dell'ultimo anno è solo presunto.

Le lire riferite ad un anno, moltiplicate per il corrispondente coefficiente, danno le lire equivalenti per l'anno 1982. Inoltre si possono comparare due anni qualsiasi, appartenenti al periodo su indicato, tenendo conto della seguente relazione:

$$L2 = \frac{L1 - c1}{c2}$$

dove $L1$ ed $L2$ sono le lire nei due anni che si vogliono confrontare, e $c1$ e $c2$ sono i rispettivi coefficienti.

Per rendere più immediata la comprensione dei risultati, questi sono accompagnati da ideogrammi costituiti da pile di monete in numero proporzionale al capitale da rappresentare.

```

100 REM INTERESSE E SCONTO
110 REM *****
120 REM DI: SERGIO BORSANI
130 REM VERSIONE: TI BASIC
140 CALL CLEAR
150 DIM COEFF(13)
160 A$(1)="FF7F1F03"
170 A$(2)="FFFFFFFF3F"
180 A$(3)="FFFFFFFFFFFF"
190 A$(4)="FFFFFFFFFC"
200 A$(5)="FFFEF8C"
210 FOR I=1 TO 5
220 CALL CHAR(127+I,A$(I))
230 CALL CHAR(135+I,A$(I))
240 NEXT I
250 CALL CHAR(133,"FFFFFFFFFFFFFFFF")
260 CALL CHAR(141,"80601C01")
270 CALL CHAR(142,"000000036C")
280 CALL CHAR(143,"010638C")
290 CALL CHAR(144,"00000000031F7FFF")
300 CALL CHAR(145,"0000013FFFFFFFF")
310 CALL CHAR(146,"0000FFFFFFFF")
320 CALL CHAR(147,"000080FCFFFFFFFF")
330 CALL CHAR(148,"0000000C0F8FEFF")
340 CALL COLOR(13,11,1)
350 CALL COLOR(14,12,1)
360 CALL COLOR(15,12,1)
370 DEF FNN=CA*(1+R/V)^(V*T)
380 RI=6
390 CO=6
400 GOSUB 4000
410 TIT0="INTERESSE E SCONTO"
420 FOR I=1 TO LEN(TIT0)
430 L=ASC(SEG$(TIT0,I,1))
440 CALL MCHAR(12,7+I,L)
450 NEXT I
460 RI=19
470 CO=10
480 GOSUB 4000
490 RI=16
500 CO=18
510 M=4
520 GOSUB 4100
530 CALL CHAR(152,"00FFFFFFFFFFFFFF")
540 CALL CHAR(153,"00FEFEFEFEFEFE")
550 CALL CHAR(154,"00FCFCFCFCFCFC")
560 CALL CHAR(155,"00FBFBFBFBFBFB")
570 CALL CHAR(156,"00FOFOFOFOFOFO")
580 CALL CHAR(157,"00EOEOEOEOEOEO")
590 CALL CHAR(158,"00COCOCOCOCOCO")
600 CALL CHAR(159,"00B0B0B0B0B0B0")
610 CALL COLOR(16,5,1)
620 FOR T=1 TO 500
630 NEXT T
640 CALL CLEAR
650 PRINT "PREMI:":
660 PRINT "1. PER LO SCONTO":
670 PRINT "2. PER L'INTERESSE SEMPLICE":
680 PRINT "3. PER L'INTERESSE COMPOSTO":
690 PRINT "4. PER IL POTERE D'ACQUISTO":
700 CALL KEY(0,K,S)
710 IF S=0 THEN 700
720 IF (K<49)+(K>52) THEN 700
730 K=K-48
740 ON K GOTO 750,1680,2660,3460
750 CALL CLEAR
760 PRINT "SCONTO":
770 CALL MCHAR(24,3,152,28)
780 PRINT ":::::":
790 RI=16
800 CO=6
810 M=3
820 GOSUB 4100
830 RI=20
840 CO=14
850 GOSUB 4000
860 RI=17
870 CO=22
880 M=4
890 GOSUB 4100
900 PRINT TAB(4);"PREZZO SCONTO COSTO":
910 PRINT "COSA VUOI CONOSCERE?":
920 PRINT "1. PREZZO","2. SCONTO","3. COSTO","4. TASSO DI SCONTO":
930 INPUT "SCELTA n. ";N
940 PRINT
950 IF (N1)+(N>4) THEN 960 ELSE 980

```



```

960 PRINT "ERRORE! RIPROVA.":
970 GOTO 930
980 ON N GOTO 990,1160,1330,1500
990 INPUT "CONOSCI LO SCONTO? (S/N) ":RISP#
1000 IF SEG$(RISP#,1,1)<>"S" THEN 1100
1010 INPUT "SCONTO = L.":SC
1020 INPUT "CONOSCI IL COSTO? (S/N) ":RISP#
1030 IF SEG$(RISP#,1,1)<>"S" THEN 1070
1040 INPUT "COSTO = L.":C
1050 PR=SC+C
1060 GOTO 1130
1070 INPUT "TASSO DI SCONTO (%) = ":R
1080 PR=SC*100/R
1090 GOTO 1130
1100 INPUT "COSTO = L.":C
1110 INPUT "TASSO DI SCONTO (%) = ":R
1120 PR=INT(C*100/(100-R))
1130 PRINT "IL PREZZO, SENZA LO SCONTO,":E":":
1140 PRINT TAB(8);"L.":PR:
1150 GOTO 4320
1160 INPUT "CONOSCI IL PREZZO? (S/N) ":RISP#
1170 IF SEG$(RISP#,1,1)="S" THEN 1220
1180 INPUT "COSTO = L.":C
1190 INPUT "TASSO DI SCONTO (%) = ":R
1200 SC=INT(C*R/(100-R))
1210 GOTO 1300
1220 INPUT "PREZZO = L.":PR
1230 INPUT "CONOSCI IL COSTO? (S/N) ":RISP#
1240 IF SEG$(RISP#,1,1)="S" THEN 1280
1250 INPUT "TASSO DI SCONTO (%) = ":R
1260 SC=INT(PR*R/(100-R))
1270 GOTO 1300
1280 INPUT "COSTO = L.":C
1290 SC=PR-C
1300 PRINT "LO SCONTO E' DI:":
1310 PRINT TAB(8);"L.":SC:
1320 GOTO 4320
1330 INPUT "CONOSCI IL PREZZO? (S/N) ":RISP#
1340 IF SEG$(RISP#,1,1)="S" THEN 1390
1350 INPUT "SCONTO = L.":SC
1360 INPUT "TASSO DI SCONTO (%) = ":R
1370 C=INT(SC*(100-R)/R)
1380 GOTO 1470
1390 INPUT "PREZZO = L.":PR
1400 INPUT "CONOSCI LO SCONTO? (S/N) ":RISP#
1410 IF SEG$(RISP#,1,1)="S" THEN 1450
1420 INPUT "TASSO DI SCONTO (%) = ":R
1430 C=INT(PR*(100-R)/100)
1440 GOTO 1470
1450 INPUT "SCONTO = L.":SC
1460 C=PR-SC
1470 PRINT "IL COSTO E' DI:":
1480 PRINT TAB(8);"L.":C:
1490 GOTO 4320
1500 INPUT "CONOSCI IL PREZZO? (S/N) ":RISP#
1510 IF SEG$(RISP#,1,1)="S" THEN 1560
1520 INPUT "SCONTO = L.":SC
1530 INPUT "COSTO = L.":C
1540 R=INT(100*SC/(SC+C))
1550 GOTO 1630
1560 INPUT "PREZZO = L.":PR
1570 INPUT "CONOSCI LO SCONTO? (S/N) ":RISP#
1580 IF SEG$(RISP#,1,1)="S" THEN 1620
1590 INPUT "COSTO = L.":C
1600 R=INT(10*(100-C*100/PR))/10
1610 GOTO 1630
1620 INPUT "SCONTO = L.":SC
1630 R=INT(1000*SC/PR)/10
1640 PRINT
1650 PRINT "IL TASSO DI SCONTO E' IL:":
1660 PRINT TAB(8);R;"%":
1670 GOTO 4320
1680 CALL CLEAR
1690 PRINT "INTERESSE SEMPLICE":
1700 CALL MCHAR(24,3,152,28)
1710 PRINT "::::::::::::":
1720 RI=17
1730 CO=6
1740 M=4
1750 GOSUB 4100
1760 RI=20
1770 CO=14
1780 GOSUB 4000
1790 RI=16
1800 CO=22
1810 M=3

```

```

1820 GOSUB 4100
1830 PRINT "CAPITALE INTERESSE MONTANTE":
1840 PRINT "PIU' RESTANDO QUESTA TERMINO-": "LOGIA IL PROGRAMMA PUO' ES-"
1850 PRINT "SERE UTILIZZATO IN TUTTI I": "PROBLEMI NEI QUALI C'E' UN"
1860 PRINT "AUMENTO PERCENTUALE":
1870 PRINT "COSA VUOI CONDSCERE?":
1880 PRINT "1. CAPITALE": "2. INTERESSE": "3. MONTANTE": "4. TASSO D'INTERESSE"
1890 INPUT "SCELTA n. ": N
1900 PRINT
1910 IF (N<1)+(N>4) THEN 1920 ELSE 1940
1920 PRINT "ERRORE! RIPROVA.":
1930 GOTO 1890
1940 ON N GOTO 1950, 2130, 2310, 2490
1950 INPUT "CONOSCI L' INTERESSE?(S/N)": RISP#
1960 IF SEG$(RISP#,1,1)="S" THEN 2010
1970 INPUT "MONTANTE = L.": MO
1980 INPUT "TASSO D'INTERESSE (%) = ": R
1990 CA=INT(MO*100/(100+R))
2000 GOTO 2090
2010 INPUT "INTERESSE = L.": IN
2020 INPUT "CONOSCI IL MONTANTE?(S/N)": RISP#
2030 IF SEG$(RISP#,1,1)="S" THEN 2070
2040 INPUT "TASSO D'INTERESSE = L.": R
2050 CA=INT(IN*100/R)
2060 GOTO 2090
2070 INPUT "MONTANTE = L.": MO
2080 CA=MO-IN
2090 PRINT
2100 PRINT "IL CAPITALE INIZIALE E' DI:":
2110 PRINT TAB(8); "L.": CA:
2120 GOTO 4320
2130 INPUT "CONOSCI IL CAPITALE?(S/N)": RISP#
2140 IF SEG$(RISP#,1,1)="S" THEN 2190
2150 INPUT "MONTANTE = L.": MO
2160 INPUT "TASSO D'INTERESSE (%) = ": R
2170 IN=INT(MO*R/(100+R))
2180 GOTO 2270
2190 INPUT "CAPITALE = L.": CA
2200 INPUT "CONOSCI IL MONTANTE?(S/N)": RISP#
2210 IF SEG$(RISP#,1,1)="S" THEN 2250
2220 INPUT "TASSO D'INTERESSE (%) = ": R
2230 IN=INT(CA*R/100)
2240 GOTO 2270
2250 INPUT "MONTANTE = L.": MO
2260 IN=INT(MO*100/(CA+MO))
2270 PRINT
2280 PRINT "L' INTERESSE E' DI:":
2290 PRINT TAB(8); "L.": IN:
2300 GOTO 4320
2310 INPUT "CONOSCI IL CAPITALE?(S/N)": RISP#
2320 IF SEG$(RISP#,1,1)="S" THEN 2370
2330 INPUT "INTERESSE = L.": IN
2340 INPUT "TASSO D'INTERESSE (%) = ": R
2350 MO=INT(IN*(100+R)/R)
2360 GOTO 2450
2370 INPUT "CAPITALE = L.": CA
2380 INPUT "CONOSCI L' INTERESSE?(S/N)": RISP#
2390 IF SEG$(RISP#,1,1)="S" THEN 2430
2400 INPUT "TASSO D'INTERESSE (%) = ": R
2410 MO=INT(CA*(100+R)/100)
2420 GOTO 2450
2430 INPUT "INTERESSE = L.": IN
2440 MO=CA+IN
2450 PRINT
2460 PRINT "IL MONTANTE E' DI:":
2470 PRINT TAB(8); "L.": MO:
2480 GOTO 4320
2490 INPUT "CONOSCI IL CAPITALE?(S/N)": RISP#
2500 IF SEG$(RISP#,1,1)="S" THEN 2550
2510 INPUT "INTERESSE = L.": IN
2520 INPUT "MONTANTE = L.": MO
2530 R=INT(1000*I/(M-I))/10
2540 GOTO 2630
2550 INPUT "CAPITALE = L.": CA
2560 INPUT "CONOSCI L' INTERESSE?(S/N)": RISP#
2570 IF SEG$(RISP#,1,1)="S" THEN 2610
2580 INPUT "MONTANTE = L.": MO
2590 R=INT(1000*MO/CA)/10-100
2600 GOTO 2630
2610 INPUT "INTERESSE = L.": IN
2620 R=INT(1000*IN/CA)/10
2630 PRINT "IL TASSO D'INTERESSE E' DEL:":
2640 PRINT TAB(8); R; "%":
2650 GOTO 4320
2660 CALL CLEAR
2670 PRINT "INTERESSE COMPOSTO":

```

```

890 CALL HCHAR(24,3,152,28)
891 PRINT :::::::::::
892 R1=20
893 CO=3
894 GOSUB 4000
895 R1=19
896 CO=10
897 M=2
898 GOSUB 4100
899 R1=18
900 CO=17
901 M=3
902 GOSUB 4100
903 R1=16
904 CO=24
905 M=5
906 GOSUB 4100
907 PRINT "CAPITALE" TAB(21) "MONTANTE":
908 INPUT "CAPITALE = L.":CA
909 INPUT "TASSO D'INTERESSE (%) = ":R
910 R=R/100
911 INPUT "GIORNI = ":B
912 INPUT "MESI = ":M
913 INPUT "ANNI = ":T
914 T=T-M/12+B/365
915 INPUT "QUANTE VOLTE ALL'ANNO VIENE COMPOSTO L'INTERESSE? ":V
916 MD=FNN
917 PRINT
918 PRINT " AL TERMINE DEL PERIODO SU": "INDICATO, IL CAPITALE E' DI-"
919 PRINT "VENTATO:":
920 PRINT TAB(6) "L.":INT(MD):
921 PRINT "VUOI IL GRAFICO ANNO PER": "ANNO? (S/N)"
922 CALL KEY(0,K,S)
923 IF S=0 THEN 3000
924 IF K=83 THEN 3050
925 IF K=78 THEN 4320
926 GOTO 3000
927 PRINT
928 INPUT "SCRIVI L'ANNO INIZIALE: ":INIZ
929 PRINT
930 IF LEN(STR$(INIZ))>4 THEN 3090 ELSE 3110
931 PRINT "ERRORE! RIPROVA":
932 GOTO 3060
933 CALL CLEAR
934 U=MD/24
935 FINE=T
936 FOR T=0 TO INT(FINE)
937 RIP=INT(FNN/U)
938 RESTO=FNN-RIP*U
939 RESTO=INT(RESTO*B/U)
940 PRINT STR$(INIZ+T)
941 CALL HCHAR(23,8,152,RIP)
942 IF RESTO=0 THEN 3220
943 CALL HCHAR(23,8+RIP,160-RESTO)
944 NEXT T
945 IF FINE=INT(FINE) THEN 3310
946 RIP=INT(MD/U)
947 RESTO=MD-RIP*U
948 RESTO=INT(RESTO*B/U)
949 PRINT STR$(INIZ+T)
950 CALL HCHAR(23,8,152,RIP)
951 IF RESTO=0 THEN 3310
952 CALL HCHAR(23,8+RIP,160-RESTO)
953 PRINT
954 PRINT "VUOI LA TABELLA? (S/N)"
955 CALL KEY(0,K,S)
956 IF S=0 THEN 3330
957 IF K=83 THEN 3380
958 IF K=78 THEN 4320
959 GOTO 3330
960 CALL HCHAR(23,1,32,64)
961 FOR T=0 TO INT(FINE)
962 PRINT STR$(INIZ+T) TAB(6) "L.":INT(FNN)
963 NEXT T
964 IF FINE=INT(FINE) THEN 3440
965 PRINT STR$(INIZ+T) TAB(6) "L.":INT(MD)
966 PRINT
967 GOTO 4320
968 CALL CLEAR
969 PRINT "POTERE DI ACQUISTO":
970 CALL HCHAR(24,3,152,28)
971 PRINT ::
972 PRINT " A causa dell'inflazione il": "potere d'acquisto della lira"
973 PRINT "diminuisce ogni anno.": " Chi nel 1975 guadagnava 800"
974 PRINT "mila lire, oggi dovrebbe guadagnare 2 milioni e 780 mila"
975 PRINT "lire, per avere uno stipen-": "dio equivalente.":

```

```

3540 PRINT " Specificando i due anni da confrontare e il capitale"
3550 PRINT " relativo ad uno di essi, il programma calcola le lire e="
3560 PRINT " equivalenti, cioè aventi lo stesso potere d'acquisto.":
3570 PRINT TAB(6); "- premi un tasto -"
3580 GOSUB 3560
3590 PRINT " Bene, continuiamo.":
3600 IF FLAG=1 THEN 3670
3610 DATA 3.696,5.4,5.079,4.522,3.747,3.179,2.699
3620 DATA 2.281,2.02,1.758,1.461,1.228,1,.896
3630 FOR I=0 TO 13
3640 READ COEFF(I)
3650 NEXT I
3660 FLAG=1
3670 PRINT "GLI ANNI CONSIDERATI VANNO DAL 1970 AL 1983.":
3680 INPUT "CAPITALE DI L.":CA
3690 INPUT "DELL'ANNO? ":ANNO1
3700 PRINT
3710 IF (ANNO1<1970)+(ANNO1>1983) THEN 3720 ELSE 3740
3720 PRINT "ERRORE! NON E' COMPRESO NEL PERIODO CONSIDERATO.":
3730 GOTO 3690
3740 INPUT "CALCOLO DELLE LIRE EQUIVALENTI NELL'ANNO? ":ANNO2
3750 PRINT
3760 IF (ANNO2<1970)+(ANNO2>1983) THEN 3770 ELSE 3790
3770 PRINT "ERRORE! NON E' COMPRESO NEL PERIODO CONSIDERATO.":
3780 GOTO 3740
3790 M=INT(COEFF(ANNO2-1970)*2+.5)
3800 RI=21-M
3810 CD=8
3820 CALL CLEAR
3830 GOSUB 4100
3840 M=INT(COEFF(ANNO1-1970)*2+.5)
3850 RI=21-M
3860 CD=20
3870 GOSUB 4100
3880 PRINT TAB(6);ANNO1;TAB(18);ANNO2
3890 PRINT
3900 PRINT "L.":ICA
3910 CA=CA*COEFF(ANNO1-1970)/COEFF(ANNO2-1970)
3920 CA=INT(CA/10)*10
3930 PRINT "L.":ICA
3940 PRINT "(uguale potere di acquisto)":
3950 GOTO 4320
3960 CALL KEY(0,K,S)
3970 IF S=0 THEN 3960
3980 CALL MCHAR(23,3,32,28)
3990 RETURN
4000 R=0
4010 FOR I=1 TO 6
4020 Y=CD+I-1
4030 IF I<4 THEN 4050
4040 R=1
4050 CALL MCHAR(RI,Y,143+I-R)
4060 CALL MCHAR(RI+1,Y,135+I-R)
4070 CALL MCHAR(RI+2,Y,127+I-R)
4080 NEXT I
4090 RETURN
4100 R=0
4110 FOR I=1 TO 6
4120 IF I<4 THEN 4140
4130 R=1
4140 CALL MCHAR(RI,CD+I-1,143+I-R)
4150 CALL MCHAR(RI+1,CD+I-1,135+I-R)
4160 NEXT I
4170 RI=RI+2
4180 R2=RI+M-2
4190 FOR RIGA=R1 TO R2
4200 CALL MCHAR(RIGA,CD,141)
4210 CALL MCHAR(RIGA,CD+1,133,3)
4220 CALL MCHAR(RIGA,CD+4,142)
4230 CALL MCHAR(RIGA,CD+5,143)
4240 NEXT RIGA
4250 R=0
4260 FOR I=1 TO 6
4270 IF I<4 THEN 4290
4280 R=1
4290 CALL MCHAR(R2+1,CD+I-1,127+I-R)
4300 NEXT I
4310 RETURN
4320 PRINT TAB(4);"VUOI CONTINUARE? (S/N)"
4330 CALL KEY(0,K,S)
4340 IF S=0 THEN 4330
4350 IF K=83 THEN 640
4360 IF K=78 THEN 4380
4370 GOTO 4330
4380 CALL CLEAR
4390 END

```

Alla linea 150 viene dimensionato il vettore COEFF con i coefficienti per determinare le lire equivalenti.

Mancando l'istruzione OPTION BASE, la base è lo zero; COEFF(0) è il coefficiente del 1970, COEFF(1) quello del 1971, e così via. Da 160 a 360 si definiscono i caratteri per designare le monete. L'immagine è creata da due routine (4000 e 4100), la prima per una sola moneta, la seconda per più di una moneta. Le variabili RI, CO, M sono rispettivamente numero di riga e di colonna dell'angolo in alto a sinistra dell'immagine e numero di monete.

Il programmatore, specificando queste tre variabili prima del GOSUB, ottiene l'immagine come e dove desidera. I caratteri per il diagramma a strisce sono definiti alle linee 530-610. Il ciclo FOR-NEXT seguente ha il semplice scopo di far restare l'immagine del titolo per qualche istante sullo schermo. Le prime due sezioni prevedono il calcolo di quattro variabili, ognuna delle quali può essere trovata conoscendo almeno due delle altre; esse sono pertanto strutturate in base ai diversi dati in input che si possono avere.

Quando una domanda prevede una risposta del tipo (S/N), per non appesantire il programma, si considera affermativa la risposta S (maiuscolo) o anche SI, per esteso; tutte le altre lettere vengono accettate e considerate una negazione.

Alle righe 2890-2920 viene chiesto il tempo per calcolare un interesse composto; si può specificare la durata in giorni, mesi e anni; il tutto è poi ridotto ad anni.

Dalla riga 3110 alla 3450 c'è la stampa del grafico, anno per anno, che rappresenta un capitale al quale è applicato un interesse composto. MO è il montante o il capitale finale, U è il valore unitario di un carattere (8 pixel). Prima di cambiare il contenuto della variabile T, esso viene memorizzato in FINE (riga 3130). Per permettere la scrittura degli anni in corrispondenza delle strisce che formano il grafico, queste sono orizzontali. Viene prima stampata una parte multipla di U; RIP (linea 3150) indica le ripetizioni del carattere "pieno"; infine si stampa un carattere che rappresenta una frazione di U e che completa la striscia. RESTO (linee 3160-3170) è una variabile che indica se questo ultimo carattere deve essere lungo 1, 2, 3, ... 7 pixel. Se RESTO = 0, ovviamente non si stampa nessun altro carattere.

Ricordo ancora, per finire, che la forma IF (...) + (...) THEN... che si trova in alcune istruzioni del programma, simula l'istruzione IF... OR... THEN... che non è disponibile nel TI Basic.

La memoria impiegata è di 10,5 Kbyte.

Geometria elementare

Il numero π greco

Come è noto dalla geometria elementare il rapporto tra la circonferenza e il diametro è un numero che viene indicato con la lettera π greco. L'irrazionalità di π è stata dimostrata per la prima volta da J.H. Lambert (1728-77), tuttavia in precedenza era già noto il suo sviluppo decimale fino a qualche centinaio di cifre dopo la virgola.

Il programma fornisce due metodi per ottenere lo sviluppo decimale di π . Il primo si basa su un'espressione trovata dal matematico inglese J. Wallis. La formula consiste in una serie infinita di prodotti di numeri razionali:

$$\pi = 2 \times \left(\frac{2}{1} \times \frac{2}{3} \times \frac{4}{3} \times \frac{4}{5} \times \frac{6}{5} \times \frac{6}{7} \times \frac{8}{7} \times \dots \right)$$

Il secondo metodo si basa sulla considerazione che la misura del perimetro di un poligono inscritto, di n lati, tende alla circonferenza per $n \rightarrow \infty$. Se L_n indica la lunghezza di un lato di un poligono inscritto di n lati, allora il lato del poligono inscritto di $2n$ lati è dato dalla formula:

$$L_{2n} = \sqrt{2 - \sqrt{4 - L_n^2}}$$

Dato un cerchio di raggio unitario, il lato del quadrato inscritto è uguale

a $\sqrt{2}$; dalla formula segue che il lato di un poligono regolare di otto lati, inscritto, è:

$$L_8 = \sqrt{2 - \sqrt{2}}$$

e di sedici lati:

$$L_{16} = \sqrt{2 - \sqrt{2 + \sqrt{2}}}$$

Più in generale:

$$L_{2^n} = \sqrt{2 - \sqrt{2 + \sqrt{2 + \dots + \sqrt{2}}}}$$

dove nel secondo membro $n-1$ radici quadrate, una interna all'altra. Il perimetro è dato dal prodotto tra il numero dei lati 2^n ed L_{2^n} . Ponendo $m = n-1$ e dividendo per due si ottiene l'espressione

$$P/2 = 2^m \sqrt{2 - \sqrt{2 + \sqrt{2 + \dots + \sqrt{2}}}}$$

con m radici quadrate una interna all'altra, il cui limite è il numero π per m tendente all'infinito.

```

100 REM CALCOLO DI PI GRECO
110 REM *****
120 REM VERSIONE: TI BASIC
130 REM
140 CALL CHAR(128,"007E2424242426")
150 CALL CLEAR
160 PRINT "PREMI:":
170 PRINT "1. PER IL METODO DI WALLIS":
180 PRINT "2. PER IL METODO DEI POLIGONI":
190 PRINT "3. PER FINIRE PREMI FCTN(4) -":
200 CALL KEY(0,K,S)
210 IF S=0 THEN 200
220 IF K=49 THEN 250
230 IF K=50 THEN 540
240 GOTO 200
250 CALL CLEAR
260 PRINT "Il matematico inglese John":
270 PRINT "la seguente espressione, che":
280 PRINT "to.":
290 PRINT CHR$(128);"/2=(2/1)*(2/3)*(4/3)*(4/5)*"
300 PRINT TAB(5);"(6/5)*(6/7)*(8/7)*...":
310 GOSUB 910
320 PRINT "Il valore di Pi Greco vie-":
330 PRINT "approssimazioni.":
340 PRINT "al prodotto si aggiunge un":
350 PRINT "Per finire premere lo zero":
360 A=2
370 P=1
380 I=0
390 CALL KEY(0,K,S)
400 IF S=0 THEN 390
410 IF K=48 THEN 100
420 I=I+1
430 P=P*A/(A+1)
440 PRINT STR$(I);";";CHR$(128);";" <"12*P
450 CALL KEY(0,K,S)
460 IF S=0 THEN 450
470 IF K=48 THEN 100
480 I=I+1
490 P=P*A/(A+1)
500 PRINT STR$(I);";";CHR$(128);";" >"12*P
510 A=A+2
520 PRINT

```

```

530 GOTO 390
540 CALL CLEAR
550 PRINT " Il numero ";CHR$(128);" si può deter-"
560 PRINT "minare calcolando il rappor-": "to R tra il perimetro di un"
570 PRINT "poligono inscritto in una": "circonferenza e il diametro."
580 PRINT " Aumentando il numero dei": "lati, R tende a ";CHR$(128);".":
590 GOSUB 910
600 PRINT " Il programma calcola la": "lunghezza del lato e del pe-"
610 PRINT "rimetro di un poligono rego-": "lare avente un numero di la-"
620 PRINT "ti esprimibile come potenza": "di 2, cioè 4, 8, 16, ecc.":
630 PRINT "scrivi l'esponente di 2":
640 INPUT "N = ";N
650 PRINT
660 PRINT "Numero dei lati:":
670 PRINT "2";STR$(N);" =";2^N::
680 IF N>2 THEN 710
690 PRINT "NON ESISTE UN POLIGONO CON": "MENO DI 3 LATI!":
700 GOTO 630
710 L=SQR(2)
720 IF N=2 THEN 780
730 IF N=3 THEN 770
740 FOR V=1 TO N-3
750 L=SQR(2+L)
760 NEXT V
770 L=SQR(2-L)
780 P=2^N*L
790 PRINT "LATO =":L
800 PRINT "PERIMETRO =":P::
810 PRINT "VALORE APPROSSIMATO PER DI-": "FETTO DI PI GRECO":
820 PRINT TAB(8);CHR$(128);" =";P/2::
830 PRINT " - VUOI CONTINUARE? (S/N) -"
840 CALL KEY(0,K,S)
850 IF S=0 THEN 840
860 IF K=83 THEN 890
870 IF K=78 THEN 100
880 GOTO 840
890 CALL HCHAR(23,3,32,28)
900 GOTO 630
910 PRINT TAB(6);"-- premi un tasto --"
920 CALL KEY(0,K,S)
930 IF S=0 THEN 820
940 CALL HCHAR(23,3,32,28)
950 RETURN

```

Alla linea 140 è definita il simbolo che rappresenti π greco; segue il menù (150-190). Alla riga 360, dopo la presentazione, inizia l'algoritmo per il calcolo di π con il metodo di Wallis. A è il numeratore delle frazioni che costituiscono i termini del prodotto infinito; il denominatore sarà alternativamente A-1 e A+1. P è uguale a $\pi/2$; naturalmente il suo valore è approssimato e il grado di approssimazione aumenta via via che il programma prosegue. I è un contatore che segna le iterazioni eseguite.

Alla linea 640 inizia la sezione ove il calcolo di π è fatto in base al perimetro dei poligoni inscritti in una circonferenza. Poiché il numero dei lati di un poligono è espresso nella forma 2^n , alla linea 640 si chiede il valore di N. Alla riga 670 c'è la stampa del numero dei lati mentre nelle linee seguenti, fino alla 760, si esegue il calcolo della misura del lato e del perimetro. L'istruzione 890 ha lo scopo di cancellare l'ultima riga scritta sul video prima di continuare.

L'ultima parte (linee 910-950) contiene una routine per il messaggio -premi un tasto-.

Va aggiunto che con il metodo di Wallis si ottiene una doppia serie convergente formata da valori approssimati per eccesso e per difetto; per ottenere una approssimazione a meno di un centesimo è necessario compiere quasi duecento iterazioni. Con il metodo dei poligoni è consigliabile attribuire ad N valori crescenti: 3, 4, 5, 6,...; per N=12 l'approssima-

mazione è già inferiore ad un decimilionesimo (10^{-7}). Per $N > 12$, tuttavia, la misura del lato diventa molto piccola mentre aumentano i cicli di calcolo; il sommarsi di piccoli errori comporta un risultato che non si avvicina più indefinitamente al valore reale di π ma che si discosta maggiormente da esso.

Ecco un esempio:

RUN

```

Scrivi l'esponente di 2
N = 8
Numero dei lati:
2^8 = 256
LATO = .0245430766
PERIMETRO = 6.283027599
VALORE APPROSSIMATO PER DI-
FETTO DI PI GRECO:
3.1415138
- VUOI CONTINUARE? (S/N) -

```

Tempo di esecuzione: 3 secondi.

Il volume del cono

Il volume del cono si trova con la nota formula: $A_{\text{base}} \times h/3$. Da un punto di vista didattico è utile mostrare come si giunga ad una formula o almeno verificare in qualche modo la sua esattezza. Ad un livello elementare, dopo aver esaminato il volume del cilindro, usando dei modelli materiali si potrà osservare che un contenitore di forma conica ha una capacità tre volte inferiore di un altro, a forma cilindrica, avente la stessa base e la stessa altezza.

Seguendo un ragionamento più propriamente matematico, si possono inscrivere nel cono innumerevoli cilindri sovrapposti (figura 4a). Detta h l'altezza del cono ed r il raggio della base, ogni cilindro inscritto avrà un'altezza x , corrispondente ad una parte ennesima di h , ed un raggio $y = r \times x/h$ (vedi figura 4b).

Il volume di ogni cilindro inscritto è pertanto: $\Delta V = \pi y^2 \times \Delta x$. Con un integrale si potrebbe scrivere subito:

$$V = \int_0^h \pi y^2 dx = \frac{\pi r^2}{h^3} \times \int_0^h x^2 dx = \frac{\pi r^2 \times h}{3}$$

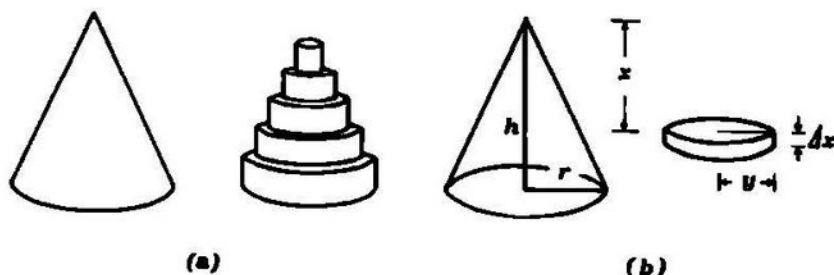


Fig. 4

ottenendo la formula nota.

Per tutto ciò, tuttavia, non è necessario un computer. Diverso è il caso in cui si desideri mostrare come all'aumentare del numero dei cilindri inscritti, o al diminuire di Δx che dir si voglia, il loro volume complessivo si avvicina indefinitamente al volume del cono.

È questo lo scopo del seguente programma.

```

100 REM *****
110 REM *
120 REM * VOLUME DEL CONO *
130 REM *
140 REM *****
150 REM DI SERGIO BORSANI
160 REM VERSIONE: TI BASIC
170 CALL CLEAR
180 LET PG=3.141592654
190 C=11
200 GOSUB 1210
210 CALL CHAR(37,"00000000FF")
220 PRINT "*****"
230 PRINT
240 PRINT "          VOLUME DEL CONO":
250 PRINT "*****":
260 FOR T=1 TO 1000
270 NEXT T
280 CALL CLEAR
290 PRINT "IL VOLUME DI UN CONO SI TRO-": "VA":
300 PRINT TAB(11); "BASE=H":
310 CALL HCHAR(23,13,37,6)
320 PRINT TAB(14); "3":
330 PRINT "PER VERIFICARE IL RISULTATO": "POSSIAMO SOMMARE I VOLUMI DI"
340 PRINT "MOLTI CILINDRI INSCRITTI.": "MAGGIORE E' IL NUMERO DEI"
350 PRINT "CILINDRI, PIU' IL LORO VOLU-": "ME COMPLESSIVO TENDE A QUEL-"
360 PRINT "LO DEL CONO.":
370 GOSUB 3610
380 CALL CLEAR
390 PRINT TAB(11); "RUNNING":
400 GOSUB 2140
410 CALL CLEAR
420 GOSUB 2850
430 CALL CHAR(91,"080B0404020201FF")
440 C=5
450 FR="OSSERVA LA FIGURA!"
460 FOR J=1 TO LEN(FR)
470 F=ASC(SEG$(FR,J,1))
480 CALL HCHAR(5,3+J,F)
490 NEXT J
500 GOSUB 1210
510 CALL HCHAR(16,10,125)
520 CALL HCHAR(16,11,124,2)
530 CALL HCHAR(16,13,91)
540 CALL HCHAR(17,11,88)
550 CALL HCHAR(9,20,158)
560 CALL HCHAR(10,20,159)
570 CALL VCHAR(11,20,127,5)
580 CALL HCHAR(16,20,125)
590 CALL HCHAR(16,21,124,2)

```

```

600 CALL HCHAR(16,23,91)
610 CALL HCHAR(15,23,156)
620 CALL HCHAR(14,22,157)
630 CALL HCHAR(13,22,156)
640 CALL HCHAR(12,21,157)
650 CALL HCHAR(11,21,156)
660 CALL HCHAR(14,17,72)
670 CALL HCHAR(14,18,45)
680 CALL HCHAR(14,19,89)
690 CALL HCHAR(17,21,88)
700 DATA 66,65,83,84,65,32,85,78,65,32,80,82,79,80,79,82,90,73,79,78,69,32
710 RESTORE 700
720 FOR J=6 TO 27
730 READ L
740 CALL HCHAR(21,J,L)
750 NEXT J
760 PR#="H:R=(H-Y):X"
770 FOR J=1 TO LEN(PR#)
780 P=ASC(SEG$(PR#,J,1))
790 CALL HCHAR(23,15+J,P)
800 NEXT J
810 PRINT " ":
820 GOSUB 3610
830 CALL CLEAR
840 PRINT "ORA CHE SAPPIAMO CALCOLARE IL RAGGIO DI OGNI CILINDRO"
850 PRINT "POSSIAMO TROVARE IL VOLUME COMPLESSIVO DEI CILINDRI."
860 PRINT
870 PRINT "SUPPONIAMO H=20 E R=10."
880 PRINT
890 DIM A(20,2)
900 K=1
910 INPUT "IN QUANTE PARTI VUOI SUDDIVIDERE L'ALTEZZA? ":A(K,1)
920 DY=20/A(K,1)
930 DEF X=Y/2
940 A(K,2)=0
950 FOR Y=20-DY TO DY STEP -DY
960 A(K,2)=A(K,2)+X*X*DY*PI
970 NEXT Y
980 PRINT
990 PRINT "N=";A(K,1);TAB(12);"V.CIL.=";INT(A(K,2)*1000)/1000
1000 PRINT
1010 INPUT "VUOI CONTINUARE? ":S$
1020 IF SEG$(S$,1,1)<>"S" THEN 1070
1030 PRINT
1040 K=K+1
1050 IF K>20 THEN 1070
1060 GOTO 910
1070 GOSUB 3320
1080 PRINT "PREMI:"
1090 PRINT
1100 PRINT "1. PER RICOMINCIARE": " "
1110 PRINT "2. PER CALCOLARE ANCORA I "
1120 PRINT "3. PER LA TABELLA DEI "
1130 PRINT "4. PER LE CONCLUSIONI": " "
1140 PRINT "5. PER FINIRE": " "
1150 INPUT L
1160 IF L=0 THEN 1190
1170 IF L>4 THEN 1190
1180 ON L GOTO 170,900,1070,3500
1190 END
1200 REM ***FIG.:CONO***
1210 CALL CHAR(128,"20100C0300000000")
1220 CALL CHAR(129,"00000000E01C0300")
1230 CALL CHAR(130,"000000000000B078")
1240 CALL CHAR(131,"0700000000000000")
1250 CALL CHAR(132,"E01F000000000000")
1260 CALL CHAR(133,"00C03F0000000000")
1270 CALL CHAR(134,"0001FE0000000000")
1280 CALL CHAR(135,"03FC000000000000")
1290 CALL CHAR(136,"F000000000000000")
1300 CALL CHAR(137,"00000000000000F")
1310 CALL CHAR(138,"0000000031CE0000")
1320 CALL CHAR(139,"0204186080000000")
1330 CALL CHAR(140,"0000B06018040201")
1340 CALL CHAR(141,"E01C030000000000")
1350 CALL CHAR(142,"000000000000F00F")
1360 CALL CHAR(143,"0000000000F0300")
1370 CALL CHAR(144,"00000000FE010000")
1380 CALL CHAR(145,"000000003FC00000")
1390 CALL CHAR(146,"0000000001FE0000")
1400 CALL CHAR(147,"0000000000000778")
1410 CALL CHAR(148,"B000000000000000")
1420 CALL CHAR(149,"031CE00000000000")
1430 CALL CHAR(150,"080B10132C306040")
1440 CALL CHAR(151,"0B0884641A0603FF")
1450 CALL CHAR(152,"E01C0300000000FF")

```

VOLUME: " "
RISULTATI: " "

```

1460 CALL CHAR(153,"0000010102020404")
1470 CALL CHAR(154,"080B101020204040")
1480 CALL CHAR(155,"80B0000000000000")
1490 CALL CHAR(156,"80B0404020201010")
1500 CALL CHAR(157,"080B040402020101")
1510 CALL CHAR(158,"80B0C0C0A0A09090")
1520 CALL CHAR(159,"8888848482828181")
1530 CALL CHAR(160,"80B0B0B0B0B0B0B0")
1540 CALL CHAR(161,"80B0B0B0FEB180B0")
1550 CALL CHAR(162,"80B0B0B0B0B0B0FF")
1560 CALL CHAR(163,"00000000000000FF")
1570 CALL HCHAR(19,C,128)
1580 CALL HCHAR(19,C+1,129)
1590 CALL HCHAR(19,C+2,130)
1600 CALL HCHAR(20,C+2,131)
1610 CALL HCHAR(20,C+3,132)
1620 CALL HCHAR(20,C+4,133)
1630 CALL HCHAR(20,C+5,134)
1640 CALL HCHAR(20,C+6,135)
1650 CALL HCHAR(20,C+7,136)
1660 CALL HCHAR(19,C+7,137)
1670 CALL HCHAR(19,C+8,138)
1680 CALL HCHAR(19,C+9,139)
1690 CALL HCHAR(18,C+9,140)
1700 CALL HCHAR(18,C+8,141)
1710 CALL HCHAR(17,C+7,142)
1720 CALL HCHAR(17,C+6,143)
1730 CALL HCHAR(17,C+5,144)
1740 CALL HCHAR(17,C+4,145)
1750 CALL HCHAR(17,C+3,146)
1760 CALL HCHAR(17,C+2,147)
1770 CALL HCHAR(18,C+2,148)
1780 CALL HCHAR(18,C+1,149)
1790 CALL HCHAR(18,C,150)
1800 CALL HCHAR(17,C,151)
1810 CALL HCHAR(17,C+1,152)
1820 CALL HCHAR(16,C+1,153)
1830 CALL HCHAR(15,C+1,154)
1840 CALL HCHAR(15,C+2,155)
1850 CALL HCHAR(14,C+2,156)
1860 CALL HCHAR(13,C+2,157)
1870 CALL HCHAR(13,C+3,158)
1880 CALL HCHAR(12,C+3,159)
1890 CALL HCHAR(11,C+3,160)
1900 CALL HCHAR(11,C+4,161)
1910 CALL HCHAR(10,C+4,162)
1920 CALL HCHAR(9,C+4,163)
1930 CALL HCHAR(9,C+5,164)
1940 CALL HCHAR(10,C+5,165)
1950 CALL HCHAR(11,C+6,166)
1960 CALL HCHAR(12,C+6,167)
1970 CALL HCHAR(13,C+7,168)
1980 CALL HCHAR(14,C+7,169)
1990 CALL HCHAR(15,C+8,170)
2000 CALL HCHAR(16,C+8,171)
2010 CALL HCHAR(17,C+9,172)
2020 CALL HCHAR(18,C+9,173)
2030 CALL HCHAR(9,C+5,174)
2040 CALL HCHAR(10,C+5,175)
2050 CALL VCHAR(11,C+5,176)
2060 CALL HCHAR(17,C+5,177)
2070 CALL HCHAR(18,C+5,178)
2080 CALL HCHAR(18,C+6,179)
2090 CALL HCHAR(18,C+8,180)
2100 CALL HCHAR(15,C+4,181)
2110 CALL HCHAR(19,C+6,182)
2120 RETURN
2130 REM *** CILINDRI INSCRITTI***
2140 CALL CHAR(91,"000000000F70B0F0")
2150 CALL CHAR(92,"00000000F00E010F")
2160 CALL CHAR(93,"8F80B0B0B0B0B0B0")
2170 CALL CHAR(94,"F101010101010101")
2180 CALL CHAR(95,"80B0B0B0B0B0B0F0")
2190 CALL CHAR(96,"010101010101010F")
2200 CALL CHAR(97,"00000000037C80FC")
2210 CALL CHAR(98,"0000000003E013F")
2220 CALL CHAR(99,"8180B0B0B0B0B0B0")
2230 CALL CHAR(100,"E01F000000000000")
2240 CALL CHAR(101,"07F8000000000000")
2250 CALL CHAR(102,"C101010101010101")
2260 CALL CHAR(103,"80B0B0B0B0B0B078")
2270 CALL CHAR(104,"010101010101011E")
2280 CALL CHAR(105,"000000010E70B0F0")
2290 CALL CHAR(106,"878180B0B0B0B0B0")
2300 CALL CHAR(107,"07E01F0000000000")
2310 CALL CHAR(108,"C01F00FF00000000")

```

```

2320 CALL CHAR(109,"03FC00FF00000000")
2330 CALL CHAR(110,"E007F80000000000")
2340 CALL CHAR(111,"7181010101010101")
2350 CALL CHAR(112,"00000080700E010F")
2360 CALL CHAR(113,"8080808080808070")
2370 CALL CHAR(114,"010101010101010E")
2380 CALL CHAR(115,"0000030C004080C0")
2390 CALL CHAR(116,"B08C83B080808080")
2400 CALL CHAR(117,"0E0180700E010000")
2410 CALL CHAR(118,"00E01F0000F00F00")
2420 CALL CHAR(119,"0000807F0000C03F")
2430 CALL CHAR(120,"000001FE000003FC")
2440 CALL CHAR(121,"0007F800000FF000")
2450 CALL CHAR(122,"7080010E70800000")
2460 CALL CHAR(123,"0D31C10101010101")
2470 CALL CHAR(124,"0000C0300C020103")
2480 CALL CHAR(125,"8080808080808040")
2490 CALL CHAR(126,"0101010101010102")
2500 CALL CHAR(127,"300C030000000000")
2510 CALL CHAR(128,"000080700E010000")
2520 CALL CHAR(129,"0000000000F00F00")
2530 CALL CHAR(130,"000000000000C03F")
2540 CALL CHAR(131,"00000000000003FC")
2550 CALL CHAR(132,"0000000000FF0000")
2560 CALL CHAR(133,"0000010E70800000")
2570 CALL CHAR(134,"0C30C00000000000")
2580 CALL CHAR(135,"1010FE7C3B10FE00")
2590 CALL CHAR(136,"000000C6442B10")
2600 CALL CHAR(137,"101010000000FE10")
2610 CALL CHAR(138,"3B7CFE1010000000")
2620 CALL CHAR(139,"0808080808080808")
2630 CALL CHAR(140,"0808087F3E1C087F")
2640 CALL CHAR(141,"7F081C3E7F080808")
2650 CALL CHAR(142,"0E70800000000000")
2660 CALL CHAR(143,"0000000000000FF0")
2670 CALL CHAR(144,"00000000003FC000")
2680 CALL CHAR(145,"0000000000FC0300")
2690 CALL CHAR(146,"000000000000F00F")
2700 CALL CHAR(147,"0000000000000080")
2710 CALL CHAR(148,"000000000000001")
2720 CALL CHAR(149,"700ED1000000FF00")
2730 CALL CHAR(150,"0000C0300C02FF03")
2740 CALL CHAR(151,"000000000000FF00")
2750 CALL CHAR(152,"000080700E010000")
2760 CALL CHAR(153,"0000000000F00F00")
2770 CALL CHAR(154,"000000000000C03F")
2780 CALL CHAR(155,"00000000000003FC")
2790 CALL CHAR(156,"000000000000FF00")
2800 CALL CHAR(157,"0000010E70800000")
2810 CALL CHAR(158,"00898B8BFF898B8B")
2820 CALL CHAR(159,"00000000FF000000")
2830 CALL CHAR(160,"00111910FF101911")
2840 RETURN
2850 DATA 6,11,6,12,7,11,7,12,8,11,8,12,8,10,8,13,9,10,9,11,9,12,9,13
2860 DATA 10,10,10,13,10,9,11,9,11,10,11,11,12,11,13,11,14
2870 DATA 10,14,12,9,12,14,12,8,13,8,13,9,13,10,13,11,13,12,13,13,14,13,15
2880 DATA 12,13,14,8,14,15,15,8,15,9,15,10,15,11,15,12,15,13,15,14,15,15
2890 RESTORE 2850
2900 FOR NC=91 TO 134
2910 READ RIGA,COLD
2920 CALL HCHAR(RIGA,COLD,NC)
2930 NEXT NC
2940 CALL HCHAR(5,6,141)
2950 CALL VCHAR(6,6,139,8)
2960 DATA 14,6,140,10,7,72,12,17,135,13,17,136,14,17,137,15,17,138
2970 DATA 14,18,125,15,18,127,15,19,128,15,20,129,15,21,130,15,22,131
2980 DATA 15,23,132,15,24,133,15,25,134,14,26,135,15,27,136,15,28,137,15,29,138
2990 DATA 13,20,152,13,21,153,13,22,154,13,23,155,13,24,156,13,25,157
3000 DATA 12,18,115,12,19,142,11,19,38,11,20,143,11,21,144,11,22,145
3010 DATA 11,23,146,11,24,147,12,24,148,12,25,149
3020 RESTORE 2960
3030 FOR I=1 TO 34
3040 READ RIGA,COLD,NC
3050 CALL HCHAR(RIGA,COLD,NC)
3060 NEXT I
3070 CALL HCHAR(12,22,150,2)
3080 CALL HCHAR(16,22,157)
3090 CALL HCHAR(16,23,158,2)
3100 CALL HCHAR(16,25,159)
3110 CALL HCHAR(17,24,88)
3120 DATA 89,61,72,47,78
3130 RESTORE 3120
3140 FOR I=1 TO 5
3150 READ L
3160 CALL HCHAR(8,18+I,L)
3170 NEXT I

```

```

3180 DOM1$="COME SI TROVA IL RAGGIO DI"
3190 FOR J=1 TO LEN(DOM1$)
3200 D=ASC(SEG$(DOM1$,J,1))
3210 CALL HCHAR(20,2+J,D)
3220 NEXT J
3230 DOM2$="OGNI CILINDRO?"
3240 FOR J=1 TO LEN(DOM2$)
3250 DD=ASC(SEG$(DOM2$,J,1))
3260 CALL HCHAR(21,2+J,DD)
3270 NEXT J
3280 GOSUB 3610
3290 CALL CLEAR
3300 RETURN
3310 REM ***TABELLA VOLUMI***
3320 CALL CLEAR
3330 PRINT "TABELLA DEI RISULTATI"
3340 PRINT "-----"
3350 PRINT
3360 PRINT "NUMERO          VOLUME"
3370 PRINT "SUDDIVISIONI    CILINDRI"
3380 PRINT "-----"
3390 PRINT
3400 FOR J=1 TO K
3410 PRINT A(J,1);TAB(19);INT(A(J,2)*1000)/1000
3420 NEXT J
3430 PRINT
3440 PRINT "VOLUME DEL CONO= 2094.395"
3450 PRINT ::
3460 GOSUB 3610
3470 CALL CLEAR
3480 RETURN
3490 REM ***CONCLUSIONE***
3500 CALL CLEAR
3510 PRINT "AUMENTANDO IL NUMERO DEI CILINDRI INSCRITTI IL LORO"
3520 PRINT "VOLUME COMPLESSIVO SI AVVI-"
3530 PRINT "CINA SEMPRE PIU' AL VALORE";TAB(11);"2094.395.":
3540 PRINT "VERIFICA CHE TALE VALORE RAPPRESENTA 1/3 DEL VOLUME"
3550 PRINT "DEL CILINDRO AVENTE LA STESS-"
3560 PRINT "SA BASE E LA STESSA ALTEZZA DEL CONO"
3570 PRINT ::
3580 GOSUB 3610
3590 CALL CLEAR
3600 GOTO 1080
3610 PRINT TAB(3);"(PREMI 1 PER CONTINUARE)"
3620 CALL KEY(0,KK,S)
3630 IF S=0 THEN 3620
3640 RETURN

```

Nella parte iniziale viene introdotto il problema con una grafica ad alta risoluzione. Le voluminose routine 1200 e 2130 sono adibite appunto alla creazione delle immagini del cono e del pluricilindro inscritto. Le istruzioni DATA (linee 2850-2880 e 2960-3010) contengono i numeri di riga, di colonna e di carattere per le successive CALL HCHAR.

Alla linea 910 viene chiesto in quante parti si vuole suddividere l'altezza. Il numero è memorizzato nella matrice A(K,1), dimensionata con K=20; K indica il numero di prove effettuate o, più propriamente, è il numero di record; in ogni record, il campo 1 contiene il numero di parti in cui è stata suddivisa l'altezza del cono, mentre 2 contiene il volume del pluricilindro inscritto. Alla linea 920, DY è l'ennesima parte dell'altezza del cono, mentre alla riga seguente, X è il raggio dell'i-esimo cilindro inscritto. Y è la sua distanza dal vertice. Il ciclo FOR-NEXT alle linee 950-970 calcola il volume del pluricilindro inscritto.

L'operazione può essere ripetuta suddividendo l'altezza in 10, 20, 30,... 100 o più parti. Quando non si intende proseguire oltre, i dati fino ad allora raccolti vengono tabulati e confrontati con il volume del cono (routine 3310). La sezione 3490-3600 contiene le conclusioni mentre le

ultime istruzioni sono una routine per il messaggio premi 1. per continuare.

Segue un esempio di utilizzazione. Il tempo impiegato è di 7 secondi ed aumenta proporzionalmente alle parti in cui è suddivisa l'altezza.

RUN

```

IN QUANTE PARTI VUOI SUDDI-
VIDERE L'ALTEZZA? 40
N = 40      V.CIL. = 2016.509
VUOI CONTINUARE? S
IN QUANTE PARTI VUOI SUDDI-
VIDERE L'ALTEZZA? 50
N = 50      V.CIL. = 2031.982
VUOI CONTINUARE?
  
```

IL teorema di Pitagora

Il teorema è fin troppo noto per essere trattato qui da un punto di vista matematico, in ogni caso il programma contiene una sezione in cui vengono forniti gli elementi essenziali accompagnati da qualche rappresentazione grafica. Nella parte applicativa, viene data la misura di un cateto o dell'ipotenusa dopo aver introdotto gli altri due elementi di un triangolo rettangolo. Un pregio consiste nell'elaborazione non solo del risultato ma anche dei passaggi che conducono ad esso, il tutto presentato con la più appropriata simbologia ottenuta mediante la definizione di caratteri particolari.

Il programma è completato da una sezione che permette di verificare se tre numeri formano una terna pitagorica o di creare nuove terne a piacere.

Tre numeri: a , b , c , formano una terna pitagorica semplicemente quando è soddisfatta la condizione

$$a^2 + b^2 = c^2$$

Invece per creare una terna pitagorica si possono usare le seguenti formule:

$$a = 2mn; \quad b = m^2 - n^2; \quad c = m^2 + n^2$$

dove m ed n sono due numeri arbitrari tali che sia $m > n$.

```

100 REM *** TEOREMA DI PITAGORA ***
110 REM DI: SERGIO BORSANI
120 REM VERSIONE: TI BASIC
130 CALL CLEAR
140 CALL CHAR(144,"FFFFFFFFFFFFFFFF")
150 CALL COLOR(15,9,14)
160 TIT9="TEOREMA DI PITAGORA"
170 FOR J=1 TO LEN(TIT9)
180 L=ASC(SEG$(TIT9,J,1))
190 CALL HCHAR(12,6+J,L)
200 NEXT J
210 CALL HCHAR(10,5,144,23)
220 CALL VCHAR(11,5,144,3)
230 CALL VCHAR(11,2,144,3)
240 CALL HCHAR(14,5,144,53)
250 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
260 CALL CHAR(129,"C0F0FCFFFFFFFF")
270 CALL CHAR(130,"00000000C0F0FCFF")
280 CALL CHAR(131,"1F1F3F3F7F7F7FFF")
290 CALL CHAR(132,"0101030307070F0F")
300 CALL CHAR(133,"FFFFFFFFFFFFFFFF")
310 CALL CHAR(134,"C0F0FCFFFFFFFF")
320 CALL CHAR(135,"00000000C0F0FCFF")
330 CALL CHAR(136,"FFFFFFFFEFCFCFB8")
340 CALL CHAR(137,"F0F0E0E0C0C0B0B0")
350 CALL CHAR(138,"0")
360 CALL CHAR(139,"01C1222214140B0B")
370 CALL CHAR(140,"000000000000FFB0")
380 CALL CHAR(141,"000000000000FF")
390 CALL CHAR(142,"701070407")
400 CALL CHAR(143,"00001B3C3C1B")
410 CALL COLOR(13,11,1)
420 CALL COLOR(14,5,1)
430 FOR T=1 TO 200
440 NEXT T
450 CALL CLEAR
460 PRINT "PREMI:":
470 PRINT "1. PER LE FORMULE":
480 PRINT "2. PER LE APPLICAZIONI":
490 PRINT "3. PER LE TERME PITAGORICHE":
500 PRINT "4. PER FINIRE.":
510 CALL KEY(0,K,S)
520 IF S=0 THEN 510
530 K=K-48
540 IF (K<1)+(K>4) THEN 510
550 ON K GOTO 560,1410,1930,2730
560 CALL CLEAR
570 F$="FORMULE"
580 FOR J=1 TO LEN(F$)
590 L=ASC(SEG$(F$,J,1))
600 CALL HCHAR(2,2+J,L)
610 NEXT J
620 CALL HCHAR(4,3,144,28)
630 R=12
640 C=10
650 GOSUB 2500
660 CALL COLOR(13,11,5)
670 CALL HCHAR(14,16,142)
680 CALL HCHAR(13,14,138,2)
690 CALL HCHAR(13,16,141)
700 CALL HCHAR(12,12,138,5)
710 CALL HCHAR(12,17,142)
720 CALL HCHAR(11,10,136)
730 CALL HCHAR(11,11,138,6)
740 CALL HCHAR(11,17,141)
750 CALL HCHAR(10,10,137)
760 CALL HCHAR(10,11,138,7)
770 CALL HCHAR(10,18,142)
780 CALL HCHAR(9,11,136)
790 CALL HCHAR(9,12,138,6)
800 CALL HCHAR(9,18,141)
810 CALL HCHAR(8,11,137)
820 CALL HCHAR(8,12,138,5)
830 CALL HCHAR(8,17,139)
840 CALL HCHAR(8,18,140)
850 CALL HCHAR(7,12,136)
860 CALL HCHAR(7,13,138,2)
870 CALL HCHAR(7,18,139)
880 CALL HCHAR(7,16,140)
890 CALL HCHAR(6,12,137)
900 CALL HCHAR(6,13,139)
910 CALL HCHAR(6,14,140)
920 FOR R=12 TO 14
930 CALL HCHAR(R,7,144,3)
940 NEXT R
950 FOR R=15 TO 20
960 CALL HCHAR(R,10,145,6)

```



```

970 NEXT R
980 CALL HCHAR(13,5,65)
990 CALL HCHAR(13,6,155)
1000 CALL HCHAR(21,12,66)
1010 CALL HCHAR(21,13,155)
1020 CALL HCHAR(6,16,67)
1030 CALL HCHAR(6,17,155)
1040 FOR T=1 TO 400
1050 NEXT T
1060 Q$="A"&CHR$(155)&"+"&B"&CHR$(155)&"="&C"&CHR$(155)
1070 FOR J=1 TO LEN(Q$)
1080 L=ASC(SEG$(Q$,J,1))
1090 CALL HCHAR(17,20+J,L)
1100 NEXT J
1110 CALL COLOR(12,16,7)
1120 CALL HCHAR(15,19,126,12)
1130 CALL VCHAR(16,19,126,3)
1140 CALL VCHAR(16,30,126,3)
1150 CALL HCHAR(19,19,126,12)
1160 FOR I=1 TO 30
1170 CALL COLOR(12,7,16)
1180 CALL COLOR(12,16,7)
1190 NEXT I
1200 GOSUB 2590
1210 CALL CLEAR
1220 CALL COLOR(13,9,1)
1230 R=20
1240 C=14
1250 GOSUB 2500
1260 CALL HCHAR(20,17,67)
1270 CALL HCHAR(21,12,65)
1280 PRINT TAB(14);"B";11
1290 PRINT " C="&CHR$(155);"="&A"&CHR$(155);"+"&B"&CHR$(155);
1300 PRINT " C="&CHR$(152);" A"&CHR$(155);"+"&B"&CHR$(155)
1310 GOSUB 2650
1320 PRINT " A"&CHR$(155);"="&C"&CHR$(155);"-"&B"&CHR$(155);
1330 PRINT " A="&CHR$(152);" C"&CHR$(155);"-"&B"&CHR$(155)
1340 GOSUB 2650
1350 PRINT " B"&CHR$(155);"="&C"&CHR$(155);"-"&A"&CHR$(155);
1360 PRINT " B="&CHR$(152);" C"&CHR$(155);"-"&A"&CHR$(155)
1370 GOSUB 2650
1380 PRINT
1390 GOSUB 2610
1400 IF K=83 THEN 560 ELSE 450
1410 CALL CLEAR
1420 PRINT "APPLICATIONI":
1430 CALL HCHAR(24,3,144,28)
1440 PRINT ":::::"
1450 CALL COLOR(13,5,1)
1460 R=20
1470 C=15
1480 GOSUB 2500
1490 CALL HCHAR(21,13,65)
1500 CALL HCHAR(20,18,67)
1510 PRINT TAB(15);"B";11
1520 INPUT "COSA VUOI TROVARE? ":X$
1530 PRINT
1540 IF X$="A" THEN 1590
1550 IF X$="B" THEN 1660
1560 IF X$="C" THEN 1730
1570 PRINT "ERRORE! DEVI SCEGLIERE A,B: OPPURE C, RIPETI":
1580 GOTO 1520
1590 INPUT "B = ":W2
1600 INPUT "C = ":W1
1610 IF W2<W1 THEN 1640
1620 PRINT "L'IPOTENUSA DEVE ESSERE MAGGIORE DEL CATETO! RIPETI":
1630 GOTO 1590
1640 W$="-"
1650 GOTO 1770
1660 INPUT "A = ":W2
1670 INPUT "C = ":W1
1680 IF W2<W1 THEN 1710
1690 PRINT "L'IPOTENUSA DEVE ESSERE MAGGIORE DEL CATETO! RIPETI":
1700 GOTO 1660
1710 W$="-"
1720 GOTO 1770
1730 INPUT "A = ":W1
1740 INPUT "B = ":W2
1750 W$="+"
1760 GOTO 1770
1770 PRINT
1780 Z1$=STR$(INT(100*W1*W1)/100)&W$&STR$(INT(100*W2*W2)/100)
1790 IF W$="+" THEN 1820
1800 Z2$=STR$(INT(100*(W1+W2*W2)/100)
1810 GOTO 1830
1820 Z2$=STR$(INT(100*(W1*W1+W2*W2)/100)

```

```

1830 PRINT TAB(9);CHR$(153);
1840 LL=LEN(Z10)-1
1850 GOSUB 2680
1860 PRINT "1. K1" = "1CHR$(152);Z10;" = "
1870 PRINT TAB(9);CHR$(153);
1880 LL=LEN(Z20)-1
1890 GOSUB 2680
1900 PRINT TAB(6);" = "1CHR$(152);Z20;" = "INT(100*SDR(VAL(Z20)))/100:::
1910 GOSUB 2610
1920 IF K=83 THEN 1410 ELSE 450
1930 CALL CLEAR
1940 PRINT "TERNE PITAGORICHE":
1950 CALL HCHAR(24,3,144,28)
1960 PRINT ":::
1970 PRINT "PREMI:"
1980 PRINT "1. PER VERIFICARE SE TRE
1990 PRINT "    NUMERI FORMANO UNA TERNA": PITAGORICA:::
2000 PRINT "2. PER CREARE NUOVE TERNE": PITAGORICHE:::
2010 CALL KEY(10,K1)
2020 IF K=0 THEN 2010
2030 IF K=49 THEN 2060
2040 IF K=50 THEN 2290
2050 GOTO 2010
2060 PRINT "SCRIVI I TRE NUMERI":
2070 INPUT "N1 = ":N(1)
2080 MAX=N(1)
2090 INPUT "N2 = ":N(2)
2100 INPUT "N3 = ":N(3)
2110 FOR I=2 TO 3
2120 IF N(I)<MAX THEN 2140
2130 MAX=N(I)
2140 NEXT I
2150 SQ=0
2160 FOR I=1 TO 3
2170 IF N(I)<MAX THEN 2200
2180 NQ=MAX^2
2190 GOTO 2210
2200 SQ=SQ+N(I)^2
2210 NEXT I
2220 PRINT "
2230 IF NQ=SQ THEN 2260
2240 PRINT "NO, I TRE NUMERI NON FORMANO": "UNA TERNA PITAGORICA"
2250 GOTO 2270
2260 PRINT "SI, I TRE NUMERI FORMANO UNA": "TERNA PITAGORICA"
2270 PRINT ":::
2280 GOTO 2480
2290 PRINT "SCRIVI DUE NUMERI (M>N)":
2300 INPUT "M = ":K1
2310 INPUT "N = ":K2
2320 PRINT "
2330 IF K1>K2 THEN 2360
2340 PRINT "ERRORE! DEVE ESSERE M>N.":
2350 GOTO 2300
2360 LL=6
2370 A=2*K1*K2
2380 LL=LL+LEN(STR$(A))
2390 B=K1*K1-K2*K2
2400 LL=LL+LEN(STR$(B))
2410 C=K1*K1+K2*K2
2420 LL=LL+LEN(STR$(C))
2430 PRINT "I TRE NUMERI":
2440 IF LL>26 THEN 2460
2450 PRINT TAB((28-LL)/2);
2460 PRINT A|B|C;
2470 PRINT "FORMANO UNA TERNA PITAGORICA":
2480 GOSUB 2610
2490 IF K=83 THEN 1930 ELSE 450
2500 CALL HCHAR(R,C,129)
2510 CALL HCHAR(R,C+1,130)
2520 CALL HCHAR(R+1,C,128,2)
2530 CALL HCHAR(R+1,C+1,129)
2540 CALL HCHAR(R+2,C+2,130)
2550 CALL HCHAR(R+2,C+3,128,4)
2560 CALL HCHAR(R+2,C+4,129)
2570 CALL HCHAR(R+2,C+5,130)
2580 RETURN
2590 PRINT TAB(5);"- PREMI UN TASTO -"
2600 GOTO 2620
2610 PRINT "VUOI RIPETERE? (S/N) -"
2620 CALL KEY(10,K1)
2630 IF K=0 THEN 2620
2640 RETURN
2650 CALL HCHAR(22,20,133)
2660 CALL HCHAR(22,21,154,5)
2670 RETURN
2680 FOR J=1 TO LL

```

```

2690 PRINT TAB(9+I);CHR$(154);
2700 NEXT I
2710 PRINT
2720 RETURN
2730 CALL CLEAR
2740 END

```

Nella parte iniziale si susseguono il titolo (16040), la definizione dei caratteri grafici (250-420) ed il menù (460-500). I caratteri degli insiemi numerici 13, 14 e 15, dal 128 al 145, servono alla creazione della classica figura che illustra il teorema di Pitagora. I caratteri 152, 153 e 154 costituiscono il segno di radice quadrata; il 155 è l'esponente 2.

Le linee 160-200 e 1060-1100 sono un esempio di uno dei modi in cui è possibile indirizzare una frase in una qualsiasi posizione dello schermo. Questo è particolarmente utile quando sul video ci sono già delle immagini; con una istruzione PRINT queste verrebbero sospinte verso l'alto. Alla linea 1410 inizia la sezione dedicata alle applicazioni. Con la routine 2500 si crea l'immagine di un triangolo che facilita la comprensione dei dati da introdurre. L'input dei dati è differenziato e dipende dall'elemento che si vuol trovare: un cateto o l'ipotenusa. W1 e W2 sono i due elementi noti, W\$ indica se i loro quadrati vanno sommati o sottratti. Il messaggio alle linee 1620 e 1690 è un esempio di come il programmatore debba prevedere eventuali errori; in questo caso si sarebbe ottenuto un radicando negativo. Il calcolo non presenta difficoltà, la stampa e le istruzioni ad essa connesse occupano tutte le righe dalla 1770 alla 1900. Infatti, come è già stato detto nella premessa, vengono dati anche i risultati intermedi, come se si trattasse di una soluzione di tipo scolastico. I programmi didattici dedicati all'infanzia devono curare particolarmente la forma se vogliono essere chiari e facilmente comprensibili.

Per controllare se tre numeri formano una terna pitagorica, essi vengono memorizzati nel vettore N(I), in MAX si carica il maggiore di essi. Successivamente NQ è il quadrato di MAX (linea 2180) e SQ è la somma dei quadrati degli altri due (linea 2200). Ovviamente i tre numeri formano una terna pitagorica solo se è soddisfatta la condizione $NQ = SQ$ (2230). Nelle linee 2290-2490 si crea una terna pitagorica; i tre numeri sono i capostipiti di una famiglia di infinite terne ottenibili con i loro multipli. Il calcolo di LL e la successiva PRINT TAB di linea 2450 servono a centrare la scritta che dà il risultato.

Seguono nell'ordine: 2500-2580, routine per il disegno di un triangolo rettangolo; per accedervi specificare R e C, rispettivamente numero di riga e di colonna corrispondenti al vertice in alto a sinistra del triangolo. 2590-2640, routine per inviare due differenti messaggi, il primo ottenibile con GOSUB 2590, il secondo con GOSUB 2610. 2680-2720, routine per la stampa di quella parte del disegno di radice che non è in linea con il radicando e che deve essere lunga tanto da contenerlo.

Angoli e distanze

Aristarco di Samo, contemporaneo di Euclide e conosciuto per la sua ipotesi eliocentrica, misurando l'angolo formato dalle congiungenti terra-luna e terra-sole durante una fase di mezza luna riuscì a stabilire, in modo sia pur grossolanamente approssimato, il rapporto tra le distanze della luna e del sole dalla terra, deducendo che la seconda è enormemente più grande della prima. Un altro grande geografo ed astronomo del passato, Eratostene, basandosi ancora su misure angolari, riuscì a stabilire la misura del raggio della terra con un errore del 15%. Grandi risultati per i mezzi a disposizione 2000 anni fa!

La nostra esperienza quotidiana ci induce a sottovalutare le possibilità offerte dalle misure angolari, ma appena si esce dalle pareti domestiche non è difficile rendersi conto delle difficoltà insite nella misura diretta delle distanze. Quanto è alta quella casa o quella montagna? Quanto è lontana quella casa al di là del fiume? O addirittura, quant'è lontana la luna? Senza una misura angolare non si potrebbe avere alcuna risposta. È vero che gli strumenti adatti per misurare con precisione gli angoli non sono generalmente disponibili alla maggior parte delle persone, tuttavia quello che è importante capire è il metodo da seguire piuttosto che utilizzarlo nella pratica.

Interrogiamo il nostro TI 99/4A. Le prime due sezioni del programma trattano la conversione delle misure angolari da radianti gradi decimali a gradi sessagesimali e le quattro operazioni con gli angoli; premiamo quindi il tasto 3 per le applicazioni. Come misurare l'altezza di una casa? Semplice. La parete forma con il suolo un angolo retto, quindi scegliamo il caso di un triangolo rettangolo. Supponiamo di trovarci a 10 metri dalla casa (almeno una misura diretta è necessaria) e di osservare il tetto con un angolo di 35° rispetto al piano di base. Il computer dà un'altezza di 7 metri.

Il calcolo è facile; detta h l'altezza, b la base e l'angolo, risulta $\operatorname{tg} \alpha = h/b$, da cui $h = b \times \operatorname{tg} \alpha$.

Similmente si potrebbe misurare il diametro della luna, nota la sua distanza e l'angolo sotto il quale appaiono gli estremi del disco. Con il TI 99/4A è anche possibile il procedimento inverso e calcolare l'angolo note le misure lineari.

Sotto quale angolo viene visto un regolo di un metro alla distanza di 200 km? In metri, $b = 200\,000$, $h = 1$... la risposta è: $0^\circ 0' 1.03''$. I moderni astronomi riescono a misurare angoli ancora più piccoli! La cosa è stupefacente.

Ancora possiamo conoscere la lunghezza del lato di un triangolo qualsiasi (scaleno) sapendo la misura degli altri due lati e l'angolo compreso. Si utilizza la formula seguente:

$$c = \sqrt{a^2 + b^2 - 2ab \times \cos \alpha}$$

Per eseguire la mappa di un territorio, a parte i metodi aerofotogrammetrici, ci si basa sulla triangolazione. Il principio è stato introdotto dall'olandese Snellius nel XVII secolo. Un triangolo è completamente determinato da un lato e dai due angoli adiacenti, cioè la conoscenza di questi tre elementi consente di dedurre tutti gli altri.

Interroghiamo ancora il computer selezionando il tasto 4. A quale distanza si trova la casa al di là del fiume se, fissata una base di 10 metri, la si vede sotto i due angoli di $92^\circ 0' 0''$ e $86^\circ 30' 0''$? La risposta è 381,3 m dall'estremo A e 381,78 dall'estremo B. E se una misura angolare è affetta da un errore di $30'$, qual è l'errore nel calcolo della distanza? Riproviamo con gli angoli di 92° e 86° , questa volta le misure danno un valore di circa 286 m; la differenza con il precedente è di quasi 100 m.

Nel nostro rapportatore mezzo grado sembra un'inezia. Ci si rende conto dell'estrema precisione richiesta in questo tipo di applicazioni.

Per il calcolo si ricorre al teorema dei seni. Con riferimento alla figura 5, sussiste la seguente relazione:

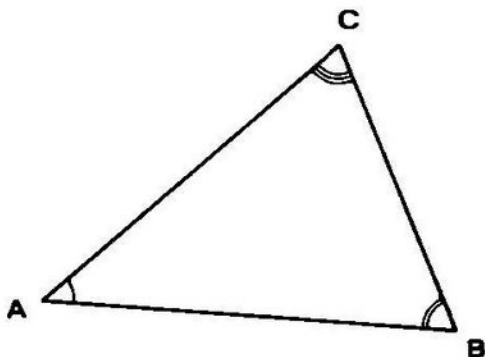


Fig. 5

$$\frac{AB}{\sin C} = \frac{BC}{\sin A} = \frac{AC}{\sin B}$$

Noti due angoli, è conosciuto anche il terzo poiché $\hat{A} + \hat{B} + \hat{C} = 180^\circ$. Se è nota la base AB, risulta:

$$BC = \frac{AB \times \sin \hat{A}}{\sin \hat{C}} \quad ; \quad AC = \frac{AB \times \sin \hat{B}}{\sin \hat{C}}$$

Come ultima applicazione viene presentato un metodo distanziometrico, usato in campo topografico, detto ad angolo parallattico variabile con stadia verticale. La stadia è un'asta graduata che generalmente va posta sul punto del quale si vuole conoscere la distanza. Essa va osservata a due differenti livelli con un ecclimetro o un teodolite, strumenti provvisti di canocchiale e cerchi goniometrici (vedi figura 6). Valgono le seguenti relazioni:

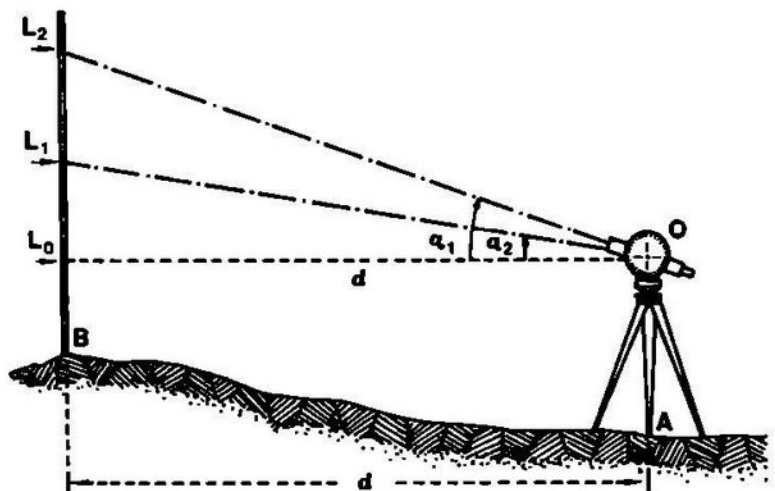


Fig. 6

$$L_0 L_1 = d \times \operatorname{tg} \alpha_2 \quad ; \quad L_0 L_2 = d \times \operatorname{tg} \alpha_1$$

Detta L la misura di $L_1 L_2$, risulta:

$$d = \frac{L}{\operatorname{tg} \alpha_1 - \operatorname{tg} \alpha_2}$$

```

100 REM  ANGOLI E DISTANZE
110 REM  *****
120 REM  di Sergio Borsani
130 REM  TI BASIC
140 CALL CLEAR
150 TIT$="ANGOLI E DISTANZE"
160 FOR I=1 TO LEN(TIT$)
170 L=ASC(SEG$(TIT$,I,1))
180 CALL HCHAR(12,7+I,L)
190 NEXT I
200 CALL CHAR(128,"OF")
210 CALL CHAR(129,"FF")
220 CALL CHAR(130,"000000000000FF")
230 CALL CHAR(131,"0000000000FF")
240 CALL CHAR(132,"0000FF")
250 CALL CHAR(133,"OFF")
260 CALL CHAR(135,"01010101010101")

```

```

270 CALL CHAR(136,"8080808080808080")
280 CALL CHAR(137,"0000000030C3FF")
290 CALL CHAR(138,"030C30C00FF20101")
300 CALL CHAR(139,"80402FF020101010")
310 CALL CHAR(141,"8080404020202020")
320 CALL CHAR(142,"00000FF0030C30C")
330 CALL CHAR(143,"0FFC30C")
340 CALL CHAR(144,"0000000030C30C")
350 CALL CHAR(145,"030C30C")
360 CALL CHAR(146,"8080404020201010")
370 CALL CHAR(147,"0808040402020101")
380 CALL CHAR(148,"C0300804030C30C")
390 CALL CHAR(149,"01010202040830C")
400 CALL CHAR(150,"000000000000FF1")
410 CALL CHAR(151,"00304B4B3")
420 CALL CHAR(152,"0024244B")
430 DEF GRA=57.295779513
440 DEF RAD=G*.01745329252
450 CALL CLEAR
460 PRINT "Premi:":
470 PRINT "1. Per le conversioni degli:" " angoli.":
480 PRINT "2. Per le operazioni con gli:" " angoli.":
490 PRINT "3. Per le applicazioni ai:" " triangoli.":
500 PRINT "4. Per la distanza di un:" " punto.":
510 PRINT "5. Per finire.":
520 CALL KEY(0,K,S)
530 IF S=0 THEN 520
540 IF (K<49)+(K>53) THEN 520
550 K=K-48
560 ON K GOSUB 640,920,1690,2490,3360
570 PRINT ":
580 PRINT "vuoi continuare con questo:" " tipo di calcolo? (s/n)"
590 CALL KEY(0,T,S)
600 IF S=0 THEN 590
610 IF (T=83)+(T=115) THEN 560
620 IF (T=78)+(T=110) THEN 450
630 GOTO 590
640 CALL CLEAR
650 PRINT "CONVERSIONI":
660 GOSUB 3240
670 PRINT "Specifica l'unita' di misura:" "nella quale e' espresso"
680 PRINT "l'angolo.":
690 PRINT "1. RADIANTI": "2. GRADI DECIMALI": "3. GRADI SESSAGESIMALI":
700 CALL KEY(0,T,S)
710 IF S=0 THEN 700
720 IF (T<49)+(T>51) THEN 700
730 ON T-48 GOTO 740,810,880
740 INPUT "RADIANTI? ":R
750 PRINT
760 PRINT R;"RAD"="";INT(1000*GRA)/1000;CHR$(151):
770 A=GRA
780 GOSUB 3340
790 PRINT "=";GRADI;CHR$(151);PRIMI;CHR$(39);SECONDI;CHR$(152)
800 GOTO 910
810 INPUT "GRADI DEC.? ":G
820 PRINT
830 A=G
840 GOSUB 3340
850 PRINT G;CHR$(151);"=";INT(1000*RAD)/1000;"RAD":
860 PRINT "=";GRADI;CHR$(151);PRIMI;CHR$(39);SECONDI;CHR$(152)
870 GOTO 910
880 GOSUB 3400
890 PRINT "=";"";INT(1000*G)/1000;CHR$(151),
900 PRINT "=";INT(1000*RAD)/1000;"RAD"
910 RETURN
920 CALL CLEAR
930 PRINT "OPERAZIONI":
940 GOSUB 3240
950 PRINT "Scegli tra le seguenti:" "operazioni":
960 PRINT "1. ADDIZIONE": "2. sottrazione"
970 PRINT "3. MOLTIPLICAZIONE": "4. DIVISIONE":
980 CALL KEY(0,T,S)
990 IF S=0 THEN 980
1000 IF (T<49)+(T>52) THEN 980
1010 ON T-48 GOTO 1020,1240,1410,1550
1020 GOSUB 3450
1030 PRINT
1040 GOSUB 3510
1050 SOMMA=0
1060 IF T=49 THEN 1130
1070 FOR I=1 TO N
1080 PRINT "angolo":IT
1090 GOSUB 3400
1100 SOMMA=SOMMA+G
1110 NEXT I
1120 GOTO 1180

```

```

1130 FOR I=1 TO N
1140 PRINT "angolo":I;
1150 INPUT "gradi? ":G
1160 SOMMA=SOMMA+G
1170 NEXT I
1180 PRINT
1190 A=SOMMA
1200 GOSUB 3340
1210 PRINT "soma =":INT(1000*SOMMA)/1000;CHR$(151);:
1220 PRINT "=:GRADI;CHR$(151);PRIMI;CHR$(39);SECONDI;CHR$(152)
1230 GOTO 1680
1240 GOSUB 3510
1250 IF T=50 THEN 1300
1260 INPUT "Angolo 1 = gradi ":G1
1270 INPUT "Angolo 2 = gradi ":G2
1280 SOMMA=G1+G2
1290 GOTO 1360
1300 PRINT "Angolo 1:"
1310 GOSUB 3400
1320 SOMMA=G
1330 PRINT "Angolo 2:"
1340 GOSUB 3400
1350 SOMMA=SOMMA+G
1360 A=SOMMA
1370 GOSUB 3340
1380 PRINT "Differenza =":INT(1000*SOMMA)/1000;CHR$(151);:
1390 PRINT "=:GRADI;CHR$(151);PRIMI;CHR$(39);SECONDI;CHR$(152)
1400 GOTO 1680
1410 GOSUB 3510
1420 IF T=50 THEN 1450
1430 INPUT "Angolo = gradi ":G
1440 GOTO 1460
1450 GOSUB 3400
1460 PRINT
1470 INPUT "moltiplicato per? ":N
1480 PRINT
1490 G=G*N
1500 A=G
1510 GOSUB 3340
1520 PRINT "Prodotto =":INT(1000*G)/1000;CHR$(151);:
1530 PRINT "=:GRADI;CHR$(151);PRIMI;CHR$(39);SECONDI;CHR$(152)
1540 GOTO 1680
1550 GOSUB 3510
1560 IF T=50 THEN 1590
1570 INPUT "Angolo = gradi ":G
1580 GOTO 1610
1590 GOSUB 3400
1600 PRINT
1610 INPUT "diviso per? ":N
1620 PRINT
1630 G=G/N
1640 A=G
1650 GOSUB 3340
1660 PRINT "Quoziente =":INT(1000*G)/1000;CHR$(151);:
1670 PRINT "=:GRADI;CHR$(151);PRIMI;CHR$(39);SECONDI;CHR$(152)
1680 RETURN
1690 CALL CLEAR
1700 PRINT "APPLICAZIONI AI TRIANGOLI":
1710 GOSUB 3240
1720 PRINT "Scegli uno dei casi seguenti":
1730 PRINT "1. Triangoli rettangoli":2. Triangoli scaleni"::
1740 CALL KEY(0,T,S)
1750 IF S=0 THEN 1740
1760 IF T=49 THEN 1790
1770 IF T=50 THEN 2250
1780 GOTO 1740
1790 PRINT ":::::":
1800 CALL MCHAR(21,11,129,10)
1810 UR=16
1820 GOSUB 3270
1830 CALL VCHAR(16,21,136,5)
1840 CALL MCHAR(20,13,141)
1850 CALL MCHAR(21,10,65)
1860 CALL MCHAR(21,21,66)
1870 CALL MCHAR(15,21,67)
1880 PRINT "(A=angolo,B=base,H=altezza)":
1890 INPUT "Cosa vuoi trovare? ":RIBPS
1900 IF RIBPS="A" THEN 1940
1910 IF RIBPS="B" THEN 2030
1920 IF RIBPS="H" THEN 2140
1930 GOTO 1890
1940 PRINT
1950 INPUT "base = ":B
1960 INPUT "altezza = ":H
1970 R=ATN(H/B)
1980 PRINT "L'angolo e' di: ":INT(1000*GRA+.5)/1000;CHR$(151);:

```



```

1990 A=GRA
2000 GOSUB 3340
2010 PRINT " " (GRADI:CHR$(151);PRIMI:CHR$(39);SECONDI:CHR$(152))
2020 GOTO 2480
2030 GOSUB 3510
2040 IF T=50 THEN 2070
2050 INPUT "angolo = gradi ":G
2060 GOTO 2080
2070 GOSUB 3400
2080 PRINT
2090 INPUT "altezza = ":H
2100 PRINT
2110 H=H/TAN(RAD)
2120 PRINT "base =":INT(100*B+.5)/100
2130 GOTO 2480
2140 GOSUB 3510
2150 IF T=50 THEN 2180
2160 INPUT "angolo = gradi ":G
2170 GOTO 2190
2180 GOSUB 3400
2190 PRINT
2200 INPUT "base = ":B
2210 PRINT
2220 H=B*TAN(RAD)
2230 PRINT "altezza =":INT(100*H+.5)/100
2240 GOTO 2480
2250 PRINT ":::::"
2260 CALL HCHAR(21,11,129,10)
2270 UR=17
2280 GOSUB 3270
2290 DATA 17,19,146,18,19,147,19,20,146,20,20,147
2300 DATA 20,13,141,21,10,65,21,21,66,18,19,67
2310 RESTORE 2290
2320 FOR I=1 TO 8
2330 READ RI,CO,NC
2340 CALL HCHAR(RI,CO,NC)
2350 NEXT I
2360 PRINT "Calcolo il lato BC, dati AB,": "BC e l'angolo compreso.":
2370 GOSUB 3510
2380 IF T=50 THEN 2410
2390 INPUT "angolo = gradi ":G
2400 GOTO 2420
2410 GOSUB 3400
2420 PRINT
2430 INPUT "lato AB = ":L1
2440 INPUT "lato AC = ":L2
2450 PRINT
2460 L=SOR(L1*L1+L2*L2-2*L1*L2*COS(RAD))
2470 PRINT "lato BC =":INT(100*L+.5)/100
2480 RETURN
2490 CALL CLEAR
2500 PRINT "DISTANZA DI UN PUNTO":
2510 GOSUB 3240
2520 PRINT "Presi:": "1. Per la triangolazione"
2530 PRINT "2. Per il metodo ad angolo": "parallattico variabile":
2540 CALL KEY(0,T,S)
2550 IF S=0 THEN 2540
2560 IF T=49 THEN 2590
2570 IF T=50 THEN 2900
2580 GOTO 2540
2590 PRINT ":::::"
2600 CALL VCHAR(18,10,135,3)
2610 UR=15
2620 GOSUB 3270
2630 FOR I=0 TO 3
2640 CALL HCHAR(17,11+I,130+I)
2650 NEXT I
2660 FOR I=0 TO 3
2670 CALL HCHAR(16,15+I,130+I)
2680 NEXT I
2690 DATA 15,19,130,15,20,131,15,21,142,15,22,143,20,11,148
2700 DATA 18,11,149,17,11,150,17,10,65,21,10,66,14,23,67
2710 RESTORE 2690
2720 FOR I=1 TO 10
2730 READ RI,CO,NC
2740 CALL HCHAR(RI,CO,NC)
2750 NEXT I
2760 INPUT "Lunghezza AB = ":L
2770 PRINT TAB(8); "A": "angolo A: "
2780 SOMMA=180
2790 GOSUB 3400
2800 SOMMA=SOMMA-G
2810 S1=SIN(RAD)
2820 PRINT TAB(8); "B": "angolo B: "
2830 GOSUB 3400
2840 S2=SIN(RAD)

```

```

2850 G=SOMMA-G
2860 S3=SIN(RAD)
2870 PRINT "Distanza AC =" ; INT(100*L*S2/S3+.5)/100
2880 PRINT "Distanza BC =" ; INT(100*L*S1/S3+.5)/100
2890 GOTO 3230
2900 PRINT "t:::":
2910 UR=15
2920 GOSUB 3270
2930 FOR I=0 TO 2
2940 CALL HCHAR(20,11+I,137+I)
2950 NEXT I
2960 CALL HCHAR(20,14,133)
2970 FOR I=0 TO 3
2980 CALL HCHAR(19,15+I,130+I)
2990 NEXT I
3000 FOR I=0 TO 3
3010 CALL HCHAR(18,19+I,130+I)
3020 NEXT I
3030 CALL VCHAR(15,23,136,3)
3040 CALL HCHAR(21,11,129,12)
3050 CALL HCHAR(21,17,68)
3060 CALL HCHAR(20,10,65)
3070 CALL HCHAR(16,24,76)
3080 PRINT "Scrivi i due angoli (A1>A2).":
3090 PRINT TAB(8);"^=";"angolo A1:"
3100 GOSUB 3400
3110 A=G
3120 S1=TAN(RAD)
3130 PRINT
3140 PRINT TAB(8);"^=";"angolo A2:"
3150 GOSUB 3400
3160 IF A>G THEN 3190
3170 PRINT "ERRORE! RIPROVA":
3180 GOTO 3080
3190 S2=TAN(RAD)
3200 PRINT
3210 INPUT "distanza L = ":L
3220 PRINT "Distanza D =" ; INT(100*L/(S1-S2)+.5)/100
3230 RETURN
3240 CALL HCHAR(24,3,95,28)
3250 PRINT "":
3260 RETURN
3270 CO=9
3280 FOR RI=20 TO UR STEP -1
3290 CO=CO+2
3300 CALL HCHAR(RI,CO,144)
3310 CALL HCHAR(RI,CO+1,145)
3320 NEXT RI
3330 RETURN
3340 GRADI=INT(A)
3350 A=(A-GRADI)*60
3360 PRIMI=INT(A)
3370 A=(A-PRIMI)*60
3380 SECONDI=INT(A*100+.5)/100
3390 RETURN
3400 PRINT "Scrivi: GRADI PRIMI SECONDI":
3410 PRINT "(" ; CHR$(151) ; " " ; CHR$(39) ; " " ; CHR$(152) ; " " ;
3420 INPUT GRADI,PRIMI,SECONDI
3430 B=GRADI+PRIMI/60+SECONDI/3600
3440 RETURN
3450 INPUT "Quanti angoli? ":N
3460 PRINT
3470 IF N<=10 THEN 3500
3480 PRINT "Spiacente, non piu' di 10.":
3490 GOTO 3450
3500 RETURN
3510 PRINT "Scegli tra: 1.gradi decimali":TAB(13);"2.gradi sessag.":
3520 CALL KEY(0,"S)
3530 IF S=0 THEN 3520
3540 IF (T<49)+(T>50) THEN 3520
3550 RETURN
3560 CALL CLEAR
3570 END

```

Dopo il titolo (140-190) sono definiti i caratteri grafici. La loro posizione nel programma non è casuale: durante il tempo impiegato dal computer per memorizzarli il titolo permane sullo schermo, fino all'apparizione del menù.

Alle linee 430-440 sono definite le funzioni per trasformare i radianti in gradi decimali e viceversa; il computer utilizza i radianti per le funzioni

trigonometriche. Il menù occupa le linee seguenti. Nelle conversioni di misura, per quanto riguarda i radianti e i gradi decimali, vengono dati i valori con l'approssimazione a meno di un millesimo.

Per le operazioni ovviamente si utilizzano i gradi decimali, il risultato è dato sia in gradi decimali che sessagesimali, cioè in gradi, primi e secondi. La variabile SOMMA la ritroviamo a linea 2780, quindi in un'altra sezione del programma, per calcolare il terzo angolo interno di un triangolo.

Le istruzioni DATA alle righe 2290-2300 e 2690-2700 contengono i numeri di riga, di colonna e di carattere per le istruzioni CALL HCHAR, evitando così di riscrivere queste parole innumerevoli volte.

Il programma utilizza alcune subroutine; vediamole una ad una. 3240-3260 scrive una linea sotto i titoli delle sezioni. 3270-3330 traccia una linea obliqua ricorrente in varie immagini; CO è il numero di colonna e RI di riga per le istruzioni CALL HCHAR, UR è l'ultima riga, la prima è sempre la numero 20, in tal modo la routine crea linee di lunghezza variabile. 3340-3390 trasforma le misure da gradi decimali a sessagesimali; "entra" la variabile A ed "escono" le variabili GRADI, PRIMI e SECONDI. 3400-3440 accetta in input una misura sessagesimale e la restituisce in forma decimale (G). 3450-3500 chiede quanti angoli devono essere sommati; il loro numero è stato posto al massimo uguale a 10. 3510-3550 chiede se per l'input seguente si desidera la forma decimale o sessagesimale.

Un'ultima considerazione. La CALL KEY di linea 520 utilizza la variabile K, essa viene fatta corrispondere al numero di sezione ($K = K-48$); se, dopo aver eseguito un calcolo, si vuole restare nella stessa sezione senza tornare al menù principale si passa direttamente alla riga 560, dove K ha sempre lo stesso valore che aveva in precedenza.

Le altre istruzioni CALL KEY non possono più usare la variabile K per non alterare il suo valore: infatti usano la variabile T.

La memoria occupata dal programma è di 8,4 Kbyte.

Ecco un esempio pratico.

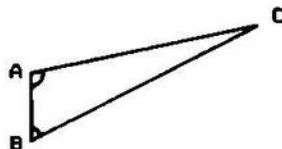
RUN

DISTANZA DI UN PUNTO

Premi:

1. Per la triangolazione
2. Per il metodo ad angolo parallattico variabile

<1>



Lunghezza AB = 10

angolo A:

scrivi: GRADI,PRIMI,SECONDI

? 92,30,15

angolo B:

scrivi: GRADI,PRIMI,SECONDI

? 86,15,10

Distanza AC = 459.98

Distanza BC = 460.52

vuoi continuare con questo
tipo di calcolo? (s/n)

Rappresentazioni grafiche

Prima di affrontare qualsiasi problema particolare è opportuno considerare brevemente le caratteristiche grafiche del T1 99/4A. L'output su video dell'home computer utilizza 24 righe e 32 colonne; pertanto è possibile individuare $24 \times 32 = 768$ posizioni nelle quali sono indirizzabili o i caratteri ASCII o altri caratteri definiti dall'utente.

Volendo stampare un asterisco alla riga 12, colonna 16, daremo l'istruzione CALL HCHAR (12,16,42), dove 42 è il numero di codice ASCII dell'asterisco.

Nella maggior parte delle applicazioni grafiche è necessario definire caratteri speciali. Per chi non avesse ancora acquisito sufficienti conoscenze sull'argomento daremo qui un esempio pratico per chiarire il procedimento.

Ogni carattere presente sul video occupa una matrice di 8×8 punti (pixel), per definire un nuovo carattere è quindi consigliabile disegnarlo prima su un foglio quadrettato sul quale si sia delimitato un quadrato formato da 8×8 quadratini. Si effettua il disegno annerendo solo i quadratini che interessano. Diviso a metà l'intero quadrato con una linea verticale, esso risulta scomposto in 16 blocchi di quattro punti ciascuno (vedi figura 7). I blocchi vanno letti, come di consueto, da sinistra a destra e dall'alto verso il basso. Dal confronto di ogni blocco con quelli della tabella di pagina 84 del manuale è possibile associare ad ognuno di essi un valore esadecimale.

Il nuovo carattere viene pertanto definito da una stringa di 16 caratteri in codice esadecimale. Nell'esempio riportato la stringa è

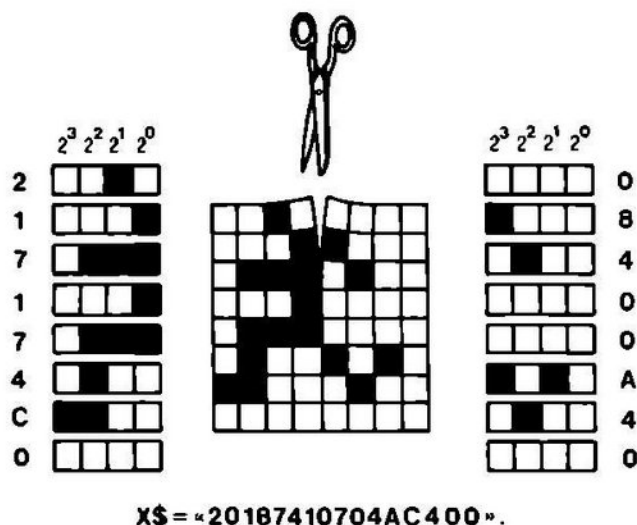


Fig. 7

A\$ = "20187410704AC400" e l'istruzione per definire il carattere è CALL CHAR(N,A\$), dove N è un numero compreso tra 32 e 159. Va ricordato che usando i numeri inferiori a 128 si ridefiniscono i caratteri ASCII, che pertanto non sono più disponibili nello stesso programma.

Alcune applicazioni attinenti alla grafica consistono nel costruire curve che rappresentano funzioni matematiche. Le curve vanno sempre costruite per punti ed è ovvio che più questi sono numerosi e ravvicinati più definita apparirà l'immagine. I migliori risultati si ottengono indirizzando i singoli pixel. Il TI 99/4A, tuttavia, non consente l'indirizzo indipendente dei singoli punti-video ma questi devono prima essere riuniti a formare un carattere ed è poi questo ad essere stampato sullo schermo. In realtà è possibile creare una routine in TI Basic per la ricerca dei punti-video che formano una curva, il loro raggruppamento in blocchi e da questi la definizione e la stampa dei caratteri. L'esecuzione è tuttavia molto lenta ed il computer impiega 3 o 4 minuti per creare il grafico di una funzione in questo modo.

Il metodo più semplice per creare l'immagine della curva consiste nello stampare una serie di asterischi. Volendo, ad esempio, disegnare una retta si inserisce nel programma l'istruzione DEF Y = A * X + B, con degli INPUT si attribuiscono i valori di A e di B, si danno alla X successivamente i valori da 1 a 32 e, se Y non è minore di 1 o maggiore di 24, si stampa l'asterisco con l'istruzione CALL HCHAR(25-Y,X,42). Le righe sono numerate dall'alto al basso e per spostare l'origine degli assi cartesiani

nell'angolo del video in basso a sinistra bisogna che il numero di riga sia 25-Y.

Poiché il TI 99/4A dispone solo di 32 colonne, la retta ottenuta con gli asterischi risulta piuttosto rudimentale e viene lecito chiedersi se non sia possibile fare di meglio.

Un buon compromesso tra l'eccessiva lentezza del programma Basic in grado di dare una curva in pixel e quello che utilizza gli asterischi consiste nel definire preventivamente un certo numero di caratteri speciali e poi far stampare quelli che più si avvicinano al tratto di curva considerato.

A titolo esplicativo consideriamo una routine adottata in un programma presente in questo capitolo.

Per raggiungere una maggiore risoluzione grafica, si simula una condizione in cui è disponibile un numero doppio di colonne ed un numero quattro volte più grande di righe.

Per ottenere un simile risultato si definiscono caratteri idealmente scomposti in due semicolonne; in ogni semicolonna si colloca un punto posto a quattro diverse altezze.

La figura 8 mostra alcuni di questi caratteri.

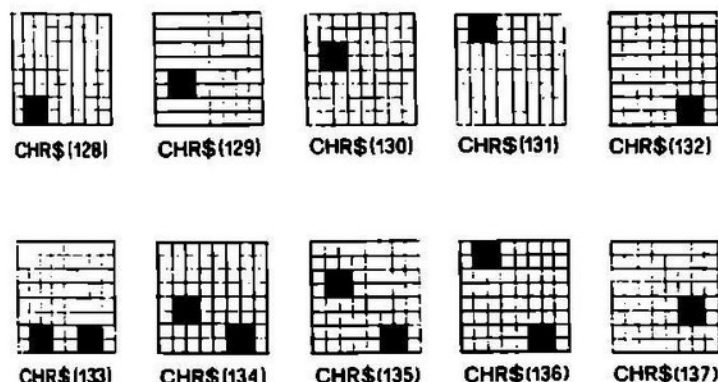


Fig. 8

Tenendo conto del fatto che potrebbe essere presente un punto in una semicolonna e mancare nell'altra, 24 caratteri sono sufficienti a coprire tutte le combinazioni. Nel nostro caso essi sono stati numerati dal 128 al 151.

Ponendo l'origine al centro dello schermo (colonna 16), si calcolano i valori della funzione nell'intervallo -12, +12 attribuendo alla variabile indipendente X valori con incremento di 0.5. Precisamente si definisce una variabile XX tale che il numero di colonne di stampa sia 16+XX;

XX assume valori interi da -12 a +12, la X è uguale alternativamente a $XX-0.75$ e $XX-0.25$. In questo modo si calcola il valore della funzione prima in corrispondenza della semicolonna sinistra e poi di quella destra. Individuate le righe di stampa il programma deve scegliere i caratteri che più si avvicinano alla funzione da rappresentare. Non va dimenticato che i punti nelle due semicolonne non si troveranno necessariamente sulla stessa riga. Si calcola pertanto $DIFF = Y - INT(Y)$ ed in base a tale differenza si trova $P1 = INT(DIFF \cdot 4) + 1$ che indica a quale livello si trova il punto all'interno del carattere e si stampa il primo carattere con l'istruzione `CALL HCHAR (13-RIGA1,16+XX,127+P1)`.

Poiché P1 assume valori interi da 1 a 4, il carattere stampato avrà un numero variabile da 128 a 131 e sarà di quelli aventi un solo punto nella semicolonna di sinistra. Successivamente, in modo analogo, si stampa il punto sulla semicolonna di destra. Si calcola ancora la differenza $DIFF = Y - INT(Y)$, si trova $P2 = (INT(DIFF \cdot 4) + 1) \cdot 5$, e se $RIGA1 = RIGA2$, si stampa il carattere con l'istruzione `CALL HCHAR(13-RIGA2,16+XX,127+P1+P2)`, altrimenti, se le righe di stampa sono diverse, si dà la stessa istruzione dopo aver posto $P1=0$.

È importante notare che la somma $127 + P1 + P2$ restituisce proprio il numero del carattere, costituito da uno o più punti, che meglio rappresenta la funzione considerata.

Il listato di questa routine è presente all'interno del programma sullo studio analitico delle parabole in questo stesso capitolo. Riportiamo invece, qui di seguito, il diagramma a blocchi.

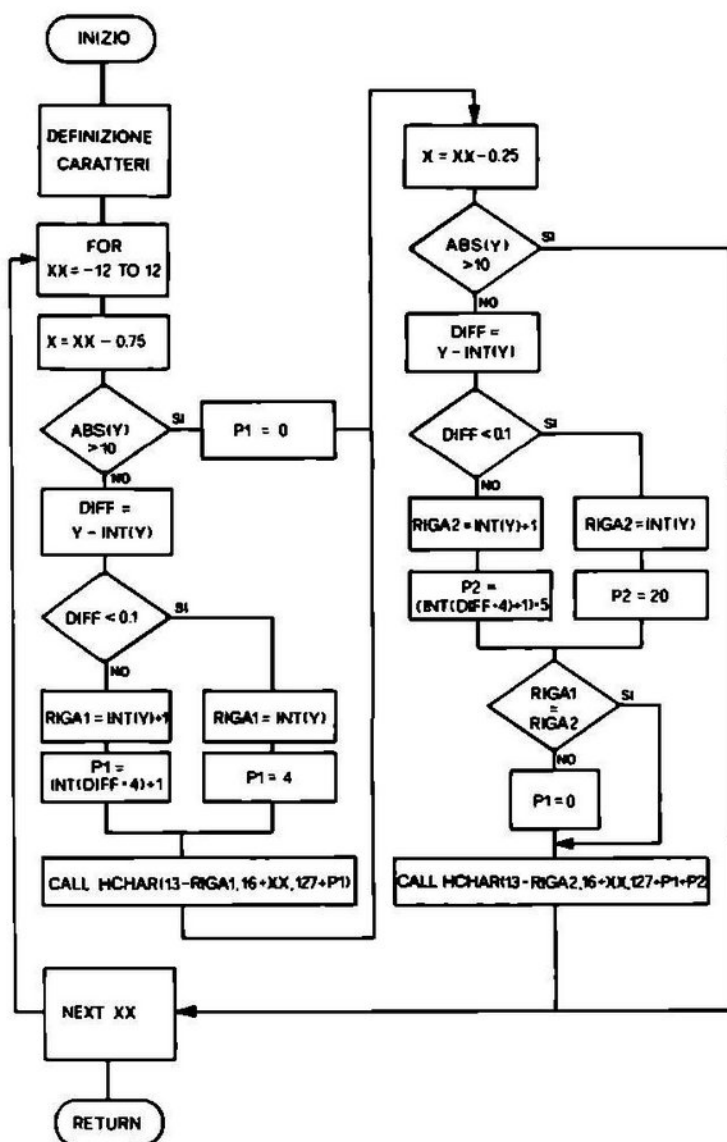
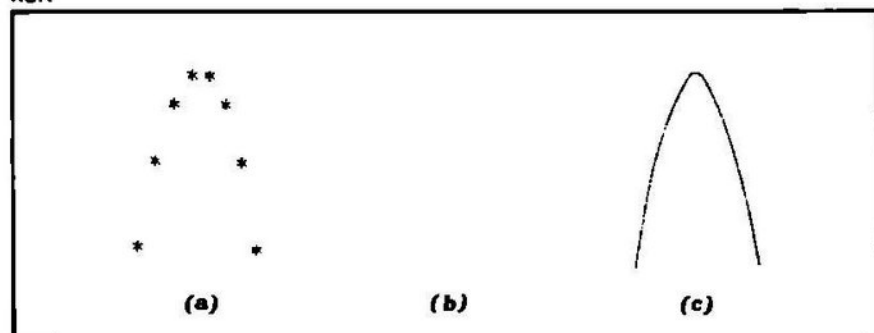


Fig. 9

Per meglio valutare i risultati ottenibili con metodi diversi, nella figura 10 viene mostrata la stessa parabola ottenuta con la stampa di asterischi (a), con la routine precedentemente esposta (b) e in alta risoluzione (c).

RUN



Un'ultima considerazione, prima di passare ai programmi. In TI Basic, quando si stampa un carattere in corrispondenza di una riga e di una colonna, esso sostituisce ogni altro carattere eventualmente presente nella stessa posizione. In alcuni casi questa caratteristica può dar luogo a qualche inconveniente. Ad esempio, se i punti che costituiscono il grafico della funzione cadono in prossimità degli assi cartesiani o se si vuole rappresentare l'intersezione tra due rette, può accadere che, per la sovrapposizione di nuovi caratteri, venga cancellata una parte del grafico precedentemente stampata. Ovviamente il fatto non si verificherebbe se si potesse indirizzare, in modo indipendente, ogni singolo pixel piuttosto che un carattere che è formato da 64 pixel.

Il problema può essere risolto con una routine che esegua la "somma" di due caratteri cioè che, dati due caratteri definiti dalle stringhe A\$ e B\$, costruisca un terzo carattere C\$ sommando, uno ad uno, i caratteri esadecimali di A\$ e B\$. Il terzo carattere porterà in posizione ON tutti i pixel del primo e tutti i pixel del secondo. Il metodo più semplice consiste nel memorizzare, con istruzioni DATA, una tabella di composizione per "sommare" i caratteri esadecimali.

Il termine sommare è posto tra virgolette perché ha un significato più generale di quello comunemente accettato. Ad esempio, il carattere esadecimale C indica un "blocco" di quattro pixel con i primi due pixel di sinistra in posizione ON, mentre il carattere esadecimale 9 significa che sono ON il primo e il quarto pixel; la loro somma è un blocco con il primo, il secondo ed il quarto pixel in posizione ON, pertanto: $C+9=D$. Analogamente $5+2=7$, ma attenti, $5+4=5$, $3+2=3$, ecc. Inoltre, F più qualsiasi altro carattere è uguale ad F. Segue la tabella di composizione completa.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	1	3	3	5	6	7	7	9	9	8	8	D	D	F	F
2	2	3	2	3	6	7	6	7	A	B	A	B	E	F	E	F
3	3	3	3	3	7	7	7	7	B	B	B	B	F	F	F	F
4	4	5	6	7	4	5	6	7	C	D	E	F	C	D	E	F
5	5	5	7	7	6	6	7	7	D	D	F	F	D	D	F	F
6	6	7	6	7	6	7	6	7	E	F	E	F	E	F	E	F
7	7	7	7	7	7	7	7	7	F	F	F	F	F	F	F	F
8	8	9	A	B	C	D	E	F	8	9	A	B	C	D	E	F
9	9	9	B	B	D	D	F	F	9	9	B	B	D	D	F	F
A	A	B	A	B	E	F	E	F	A	B	A	B	E	F	E	F
B	B	B	B	B	F	F	F	F	B	B	B	B	F	F	F	F
C	C	D	E	F	C	D	E	F	C	D	E	F	C	D	E	F
D	D	D	F	F	D	D	F	F	D	D	F	F	D	D	F	F
E	E	F	E	F	E	F	E	F	E	F	E	F	E	F	E	F
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

Tolto l'elemento estremo F, che è costante, rimane una matrice $S(15,15)$. Letti singolarmente, i caratteri esadecimali con l'istruzione $SEG\%$ si trasformano in numeri, pertanto $A=10$, $B=11$, ecc. e si ha la loro somma richiamando l'elemento della matrice $S(N1,N2)$. La routine termina con la composizione della stringa che definisce il carattere somma.

Il programma che segue è fatto a scopo dimostrativo. Esso accetta in input due stringhe di caratteri esadecimali, stampa i due simboli ad esse associati e, a fianco, la loro "somma". Non sarà difficile adattarlo opportunamente in altri programmi attinenti alla grafica.

```

100 CALL CLEAR
110 DIM S$(15,15)
120 FOR J=0 TO 14
130 FOR K=0 TO 14
140 READ S$(J,K)
150 NEXT K
160 NEXT J
170 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E
180 DATA 1,1,3,3,5,6,7,7,9,9,8,8,D,D,F,F
190 DATA 2,3,2,3,6,7,6,7,A,B,A,B,E,F,E,F
200 DATA 3,3,3,3,7,7,7,7,B,B,B,B,F,F,F,F
210 DATA 4,5,6,7,4,5,6,7,C,D,E,F,C,D,E,F
220 DATA 5,5,7,7,6,6,7,7,D,D,F,F,D,D,F,F
230 DATA 6,7,6,7,6,7,6,7,E,F,E,F,E,F,E,F
240 DATA 7,7,7,7,7,7,7,7,F,F,F,F,F,F,F,F
250 DATA 8,9,A,B,C,D,E,F,8,9,A,B,C,D,E,F
260 DATA 9,9,B,B,D,D,F,F,9,9,B,B,D,D,F,F
270 DATA A,B,A,B,E,F,E,F,A,B,A,B,E,F,E,F
280 DATA B,B,B,B,F,F,F,F,B,B,B,B,F,F,F,F
290 DATA C,D,E,F,C,D,E,F,C,D,E,F,C,D,E,F
300 DATA D,D,F,F,D,D,F,F,D,D,F,F,D,D,F,F
310 DATA E,F,E,F,E,F,E,F,E,F,E,F,E,F,E,F
320 INPUT "STRINGA CARATTERE 1" A$
330 PRINT
340 INPUT "STRINGA CARATTERE 2" B$
350 C$=""
360 FOR I=1 TO 15

```

```

270 L1=ASC(SEG$(A$,I,1))
280 L2=ASC(SEG$(B$,I,1))
290 IF L1=80 THEN 420
400 V1=L1-48
410 GOTO 430
420 V1=L1-55
430 IF L1=60 THEN 460
440 V2=L2-48
450 GOTO 470
460 V2=L2-55
470 IF V1=15 THEN 510
480 IF V2=15 THEN 510
490 C$=C$%$$(V1,V2)
500 GOTO 520
510 C$=C$%$F"
520 NEXT I
530 C$=SEG$(C$,2,16)
540 CALL CLEAR
550 CALL CHAR(128,A1)
560 CALL CHAR(129,B1)
570 CALL CHAR(130,C$)
580 CALL HCHAR(12,10,128)
590 CALL HCHAR(12,12,43)
600 CALL HCHAR(12,14,129)
610 CALL HCHAR(12,16,61)
620 CALL HCHAR(12,18,130)
630 PRINT "VUOI CONTINUARE? (S/N) "
640 CALL KEY(0,K,S)
650 IF S=0 THEN 640
660 IF K=83 THEN 320
670 END

```

Una delle più comuni rappresentazioni grafiche è costituita dai diagrammi a strisce o a colonne.

In essi le strisce sono di lunghezza proporzionale ai valori numerici da rappresentare. Rispetto ad una tabella essi forniscono informazioni con molta più immediatezza.

Grafici a strisce

Questo programma permette di memorizzare dati consistenti ognuno in un elemento descrittivo ed un corrispondente valore numerico. Potrebbero essere, ad esempio, nomi di monti e rispettive altezze, anni e popolazione censita, nazioni e corrispondenti superfici in kmq, nomi degli studenti di una classe ed il loro peso, giorni del mese e temperature, ecc. La tabella ottenuta in un primo tempo può essere successivamente ordinata o in ordine alfabetico o secondo le misure crescenti. I dati vengono poi rappresentati con un diagramma a strisce (ortogramma) il cui dimensionamento è completamente automatico nel senso che, in base alle misure introdotte, il programma suddivide l'asse verticale in un certo numero di parti e, accanto, stampa i valori multipli di una opportuna unità di misura. Ad esempio, se la massima misura è 36, l'asse verticale viene diviso in 5 parti accompagnate dai valori 10, 20, 30, 40, e 50; se il massimo valore introdotto è 2800, l'asse è diviso in tre parti accompagnate dai valori 1000, 2000, 3000. Il dimensionamento automatico non è cosa difficile da ottenersi, ma un'adeguata notazione scritta delle grandezze è una cosa non comune e certamente utile per una corretta lettura del grafico.

Un'altra notevole prerogativa del programma apparirà con evidenza dalle seguenti precisazioni.

Ogni carattere del TI 99/4A occupa una matrice di 8×8 punti video (pixel). Il modo più semplice per ottenere la striscia di un ortogramma consiste nel definire un carattere con tutti i punti colorati, cosa che può essere fatta con le istruzioni

```
CALL CHAR(128,"F")
CALL COLOR(13,9,9)
```

che creano un quadratino rosso, e successivamente stampare i quadratini uno sull'altro per l'altezza desiderata. È evidente che in questo modo le colonnine differiranno per quantità discrete, pari a 8 punti video, e misure non molto diverse tra loro verranno rappresentate sul video con colonnine della medesima altezza. Per eliminare questo inconveniente sono stati definiti anche "blocchi" alti 1 pixel, 2 pixel, 3 pixel, ecc. e si è creata una routine che stampa la colonna fino ad una certa altezza utilizzando i quadratini "pieni" e poi la completa sovrapponendo il carattere più opportuno. In tal modo si utilizzano tutti i 192 pixel disponibili sull'asse verticale e si possono apprezzare visivamente anche piccole differenze tra le misure.

Sull'asse orizzontale, a scelta dell'utente, appaiono le iniziali degli elementi rappresentati, oppure i numeri progressivi, le lettere dell'alfabeto, le iniziali dei mesi dell'anno, ecc.

Al termine, se si desidera, viene calcolata la media aritmetica; il suo valore è dato in forma numerica sotto al diagramma ed in forma grafica, con una linea orizzontale, sul diagramma stesso.

```
100 REM *****
110 REM * ORTOGRAMMI *
120 REM *****
130 REM versione: TI BASIC
140 CALL CLEAR
150 OPTION BASE 1
160 DIM A$(22,2),X(22),K$(22)
170 PRINT "-----": " "
180 PRINT "-----ORTOGRAMMI"-----" "
190 PRINT "-----"
200 PRINT : " "
210 FOR TEMPO=1 TO 500
220 NEXT TEMPO
230 CALL CLEAR
240 STAM=0
250 INPUT "QUANTI SONO GLI ELEMENTI DA INSERIRE? (N.MASSIMO=22) " : Q
260 IF Q<=22 THEN 300
270 PRINT " * VALORE TROPPO ALTO!! "
280 PRINT
290 GOTO 250
300 PRINT
310 PRINT "PER OGNI ELEMENTO INSERISCI IL NOME E LA MISURA SEPARATI DA UNA VIRGOLA"
320 PRINT " " : " " " NOME, MISURA "
330 PRINT " " : " " "-----"
340 PRINT " " : " " "-----"
350 INPUT A$(1,1),A$(1,2)
360 MAX=VAL(A$(1,2))
370 SOMMA=VAL(A$(1,2))
380 FOR R=2 TO Q
390 PRINT R;
```

```

400 INPUT A$(R,1),A$(R,2)
410 SOMMA=SOMMA+VAL(A$(R,2))
420 IF MAX>VAL(A$(R,2)) THEN 440
430 MAX=VAL(A$(R,2))
440 NEXT R
450 CALL CLEAR
460 PRINT "PREMI:"
470 PRINT "1. PER ORDINARE LA TABELLA          SECONDO I NOMI"
480 PRINT "2. PER ORDINARE LA TABELLA          SECONDO LE MISURE"
490 PRINT "3. PER VEDERE LA TABELLA": "1" "2"
500 INPUT J
510 IF J=1 THEN 560
520 IF J=2 THEN 560
530 IF J=3 THEN 1010
540 PRINT " ": "QUESTO NUMERO NON E' AMMESSO"
550 GOTO 430
560 FOR R=1 TO Q
570 K$(R)=A$(R,J)
580 IF J<2 THEN 620
590 IF LEN(K$(R))>LEN(STR$(MAX)) THEN 620
600 K$(R)="0"&K$(R)
610 GOTO 590
620 NEXT R
630 GOSUB 1110
640 PRINT " ": "TABELLA ORDINATA:" "1" "2"
650 FOR R=1 TO Q
660 PRINT R;A$(X(R),1);TAB(20);A$(X(R),2)
670 NEXT R
680 PRINT
690 PRINT "PREMI 1. PER CONTINUARE"
700 CALL KEY(0,KEY,STATUS)
710 IF STATUS=0 THEN 700
720 CALL CLEAR
730 PRINT "PREMI:"
740 PRINT "1. PER VEDERE LA TABELLA"
750 PRINT "2. PER ORDINARE LA TABELLA"
760 PRINT "3. PER OTTENERE IL GRAFICO"
770 PRINT "4. PER RICOMINCIARE"
780 PRINT "5. PER FINIRE": " ": " ": " ": " "
790 PRINT
800 INPUT "SCELTA? " :RISP
810 IF RISP<1 THEN 840
820 IF RISP>5 THEN 840
830 GOTO 860
840 PRINT "NUMERO NON AMMESSO"
850 GOTO 730
860 ON RISP GOTO 1010,870,900,230,890
870 CALL CLEAR
880 GOTO 460
890 END
900 CALL CHAR(128,"FFFF000000000000")
910 CALL CHAR(129,"7F7F7F7F7F7F7F7F")
920 CALL CHAR(130,"007F7F7F7F7F7F7F")
930 CALL CHAR(131,"00007F7F7F7F7F7F")
940 CALL CHAR(132,"0000007F7F7F7F7F")
950 CALL CHAR(133,"000000007F7F7F7F")
960 CALL CHAR(134,"00000000007F7F7F")
970 CALL CHAR(135,"0000000000007F7F")
980 CALL CHAR(136,"000000000000007F")
990 CALL CHAR(139,"0701010101010101")
1000 GOTO 1240
1010 CALL CLEAR
1020 PRINT "TABELLA DEI DATI:" "1"
1030 FOR R=1 TO Q
1040 PRINT R;A$(R,1),A$(R,2)
1050 NEXT R
1060 PRINT
1070 PRINT "-- PREMI (1) PER CONTINUARE --"
1080 CALL KEY(0,KEY,STATUS)
1090 IF STATUS=0 THEN 1080
1100 GOTO 720
1110 FOR A=1 TO Q
1120 P=1
1130 FOR B=1 TO Q
1140 IF K$(A)>K$(B) THEN 1190
1150 IF K$(A)=K$(B) THEN 1170
1160 GOTO 1200
1170 IF A>B THEN 1190
1180 GOTO 1200
1190 LET P=P+1
1200 NEXT B
1210 X(P)=A
1220 NEXT A
1230 RETURN
1240 REM ***DIMENSIONAMENTO
1250 GOSUB 2080

```

```

1260 MAX1=MAX
1270 MDLT=1
1280 IF MAX1<9999 THEN 1320
1290 MAX1=MAX1/1000
1300 MDLT=MDLT*1000
1310 GOTO 1280
1320 VALUN1=MAX1/18
1330 E=1
1340 IF VALUN1<.1#E THEN 1400
1350 IF VALUN1<.2#E THEN 1750
1360 IF VALUN1<.333#E THEN 1850
1370 IF VALUN1<.666#E THEN 1950
1380 E=E*10
1390 GOTO 1340
1400 IF E<1 THEN 1540
1410 CALL HCHAR(20,4,48)
1420 CALL HCHAR(15,7,139)
1430 CALL HCHAR(15,6,53)
1440 CALL HCHAR(15,5,44)
1450 CALL HCHAR(15,4,48)
1460 CALL HCHAR(10,7,139)
1470 CALL HCHAR(10,4,49)
1480 CALL HCHAR(5,7,139)
1490 CALL HCHAR(5,6,53)
1500 CALL HCHAR(5,5,44)
1510 CALL HCHAR(5,4,49)
1520 COEFF=.1
1530 GOTO 2260
1540 CALL HCHAR(20,6,48)
1550 G1$=STR$(.5#E)&CHR$(139)
1560 G2$=STR$(E)&CHR$(139)
1570 G3$=STR$(1.5#E)&CHR$(139)
1580 FOR I=1 TO LEN(G1$)
1590 G=ASC(SEG$(G1$,I,1))
1600 COL=7-LEN(G1$)+I
1610 CALL HCHAR(15,COL,G)
1620 NEXT I
1630 FOR I=1 TO LEN(G2$)
1640 G=ASC(SEG$(G2$,I,1))
1650 COL=7-LEN(G2$)+I
1660 CALL HCHAR(10,COL,G)
1670 NEXT I
1680 FOR I=1 TO LEN(G3$)
1690 G=ASC(SEG$(G3$,I,1))
1700 COL=7-LEN(G3$)+I
1710 CALL HCHAR(5,COL,G)
1720 NEXT I
1730 COEFF=.1#E
1740 GOTO 2260
1750 CALL HCHAR(20,6,48)
1760 FOR I=15 TO 5 STEP -5
1770 CALL HCHAR(I,7,139)
1780 W=(20-I)/5
1790 CALL HCHAR(I,7-LEN(STR$(E)),48+W)
1800 IF E=1 THEN 1820
1810 CALL HCHAR(I,8-LEN(STR$(E)),48,LEN(STR$(E))-1)
1820 NEXT I
1830 COEFF=.2#E
1840 GOTO 2260
1850 CALL HCHAR(20,6,48)
1860 FOR I=17 TO 5 STEP -3
1870 CALL HCHAR(I,7,139)
1880 W=(20-I)/3
1890 CALL HCHAR(I,7-LEN(STR$(E)),48+W)
1900 IF E=1 THEN 1920
1910 CALL HCHAR(I,8-LEN(STR$(E)),48,LEN(STR$(E))-1)
1920 NEXT I
1930 COEFF=.333#E
1940 GOTO 2260
1950 CALL HCHAR(20,6,48)
1960 FOR I=17 TO 8 STEP -3
1970 CALL HCHAR(I,7,139)
1980 W=(20-I)/1.5
1990 CALL HCHAR(I,7-LEN(STR$(E)),48+W)
2000 IF E=1 THEN 2020
2010 CALL HCHAR(I,8-LEN(STR$(E)),48,LEN(STR$(E))-1)
2020 NEXT I
2030 CALL HCHAR(5,7,139)
2040 CALL HCHAR(5,6-LEN(STR$(E)),49)
2050 CALL HCHAR(5,7-LEN(STR$(E)),49,LEN(STR$(E)))
2060 COEFF=.666#E
2070 GOTO 2260
2080 REM ***ASSI DI RIFERIMENTO***
2090 CALL CLEAR
2100 PRINT "QUALE GRANDEZZA VUOI RAPPRESENTARE SULL'ASSE ORIZZONTALE?"
2110 PRINT

```

```

2120 PRINT "PREMI:"
2130 PRINT "1. PER AVERE NUMERI          PROGRESSIVI"
2140 PRINT "2. PER AVERE LE INIZIALI"
2150 PRINT "3. PER AVERE LETTERE          PROGRESSIVE"
2160 PRINT "4. PER AVERE LE INIZIALI DEI MESI DELL'ANNO"
2170 PRINT "5. PER AVERE LE INIZIALI DEI GIORNI DELLA SETTIMANA:" " "
2180 INPUT B
2190 CALL CLEAR
2200 CALL CHAR(137,"0101010101010101")
2210 CALL CHAR(138,"0707000000000000")
2220 CALL HCHAR(20,7,138)
2230 CALL HCHAR(20,8,128,23)
2240 CALL VCHAR(2,7,137,18)
2250 RETURN
2260 IF MOLT=1 THEN 2320
2270 Z1$="X"&STR$(MOLT)
2280 FOR I=1 TO LEN(Z1$)
2290 Z=ASC(SEG$(Z1$,I,1))
2300 CALL HCHAR(1,3+I,Z)
2310 NEXT I
2320 Q1=0
2330 IF Q1<24 THEN 2350
2340 Q1=23
2350 ON B GOTO 2360,2540,2610,2650,2780
2360 FOR I=1 TO 9
2370 CALL HCHAR(21,7+I,48+I)
2380 NEXT I
2390 FOR I=1 TO 9
2400 CALL HCHAR(21,17+I,48+I)
2410 NEXT I
2420 CALL HCHAR(21,28,49)
2430 CALL HCHAR(21,29,50)
2440 CALL HCHAR(21,30,51)
2450 CALL CHAR(140,"0047C948484848E7")
2460 CALL CHAR(141,"000080B0B0B0B000")
2470 CALL CHAR(142,"0000010000000001")
2480 CALL CHAR(143,"00E3141424444B4F3")
2490 CALL CHAR(144,"00B0404040404080")
2500 CALL HCHAR(22,17,140)
2510 CALL HCHAR(22,18,141)
2520 CALL HCHAR(22,26,142)
2530 CALL HCHAR(22,27,143)
2540 CALL HCHAR(22,28,144)
2550 GOTO 2910
2560 FOR I=1 TO Q1
2570 INIZ=ASC(SEG$(A$(I,1),1,1))
2580 CALL HCHAR(21,7+I,INIZ)
2590 NEXT I
2600 BOTO 2910
2610 FOR I=1 TO Q1
2620 CALL HCHAR(21,7+I,64+I)
2630 NEXT I
2640 BOTO 2910
2650 CALL HCHAR(21,8,71)
2660 CALL HCHAR(21,9,70)
2670 CALL HCHAR(21,10,77)
2680 CALL HCHAR(21,11,65)
2690 CALL HCHAR(21,12,77)
2700 CALL HCHAR(21,13,71)
2710 CALL HCHAR(21,14,76)
2720 CALL HCHAR(21,15,65)
2730 CALL HCHAR(21,16,83)
2740 CALL HCHAR(21,17,79)
2750 CALL HCHAR(21,18,78)
2760 CALL HCHAR(21,19,68)
2770 GOTO 2910
2780 COL=0
2790 H=1
2800 CALL HCHAR(21,COL,76)
2810 CALL HCHAR(21,COL+1,77)
2820 CALL HCHAR(21,COL+2,77)
2830 CALL HCHAR(21,COL+3,71)
2840 CALL HCHAR(21,COL+4,86)
2850 CALL HCHAR(21,COL+5,83)
2860 CALL HCHAR(21,COL+6,68)
2870 COL=COL+7
2880 H=H+1
2890 IF H>3 THEN 2910
2900 GOTO 2800
2910 REM ***GRAFICO***
2920 FOR I=1 TO Q1
2930 NUMRIGHE=INT(VAL(A$(I,2))/(COEFF*MOLT))
2940 FOR J=19 TO 20-NUMRIGHE STEP -1
2950 CALL HCHAR(J,7+I,129)
2960 NEXT J
2970 DIFFERENZA=VAL(A$(I,2))/(COEFF*MOLT)-NUMRIGHE

```



```

2980 PDEC=INT(B*DIFFERENZA+1)
2990 IF PDEC=1 THEN 3010
3000 CALL HCHAR(19-NUMRIGHE,7+I,138-PDEC)
3010 NEXT I
3020 IF STAM=1 THEN 3070
3030 GOSUB 3150
3040 GOSUB 3340
3050 STAM=1
3060 GOTO 2910
3070 FRASE$=" PREMI (1) PER CONTINUARE -"
3080 FOR I=1 TO LEN(FRASE$)
3090 NI=ASC(SEG$(FRASE$,I,1))
3100 CALL HCHAR(24,2+I,NI)
3110 NEXT I
3120 CALL KEY(0,KEY,STATUS)
3130 IF STATUS=0 THEN 3120
3140 GOTO 720
3150 REM *** MEDIA ***
3160 MEDIA$="VUOI IL VALORE MEDIO? (S/N)"
3170 FOR I=1 TO LEN(MEDIA$)
3180 ME=ASC(SEG$(MEDIA$,I,1))
3190 CALL HCHAR(24,2+I,ME)
3200 NEXT I
3210 CALL KEY(0,KEY,STATUS)
3220 IF STATUS=0 THEN 3210
3230 IF KEY=83 THEN 3260
3240 IF KEY=115 THEN 3260
3250 GOTO 3320
3260 VMEDIO$="VALORE MEDIO = "&STR$(INT((SOMMA+1000/Q)/1000))
3270 FOR I=1 TO LEN(VMEDIO$)
3280 VM=ASC(SEG$(VMEDIO$,I,1))
3290 CALL HCHAR(23,2+I,VM)
3300 NEXT I
3310 GOTO 3330
3320 CALL HCHAR(24,3,32,28)
3330 RETURN
3340 REM ***MEDIA SUL VIDEO
3350 CALL CHAR(150,"FF00000000000000")
3360 CALL CHAR(151,"00000000000000FF")
3370 CALL CHAR(152,"000000000000FF00")
3380 CALL CHAR(153,"0000000000FF0000")
3390 CALL CHAR(154,"00000000FF000000")
3400 CALL CHAR(155,"000000FF00000000")
3410 CALL CHAR(156,"0000FF0000000000")
3420 CALL CHAR(157,"00FF000000000000")
3430 RM=INT((SOMMA/Q)/(COEFF*MOLT))
3440 DIFF=(SOMMA/Q)/(COEFF*MOLT)-RM
3450 PDM=INT(B*DIFF+1)
3460 IF PDM<1 THEN 3490
3470 CALL HCHAR(20-RM,B,150,23)
3480 GOTO 3500
3490 CALL HCHAR(19-RM,B,149+PDM,23)
3500 RETURN

```

Alla linea 160 viene riservata un'area di memoria per le matrici ed i vettori A\$, X e K\$; serviranno per accogliere i dati in input e come indici e variabili provvisorie per il loro ordinamento. Il numero massimo degli elementi da inserire è stato posto uguale a 22 poiché le prime colonne dello schermo sono occupate da uno degli assi di riferimento e dai valori delle graduazioni. L'input dei dati avviene alla linea 340 e seguenti, contemporaneamente viene caricato il valore massimo in MAX e la somma di tutte le misure in SOMMA. La loro presenza è da mettere in relazione, rispettivamente, al dimensionamento automatico del grafico e al calcolo della media. Tutti i dati sono inseriti in forma di stringa, pertanto, per eseguire su di essi qualsiasi operazione aritmetica, bisognerà convertirli in numeri con la funzione VAL. Nella linea 490, J rappresenta la scelta sul modo in cui deve essere fatto l'ordinamento. A dire il vero il valore 3 rimanda semplicemente alla linea 1010 in cui si stampano i dati così come sono stati introdotti, altrimenti J rappresenta il campo dei record secondo il quale si vuole l'ordinamento. Pertanto, se il suo valore è

1, tutti gli elementi del primo campo saranno trasferiti nel vettore K\$; se il suo valore è 2, nel medesimo vettore saranno trasferiti tutti gli elementi del secondo campo. Per il successivo ordinamento i confronti verranno effettuati proprio tra gli elementi di questo nuovo vettore. La routine adibita a questo scopo è posta più avanti alle linee 1110-1230.

Nella istruzione PRINT di riga 610, X(R) è un vettore che funge da indice della matrice A\$ i cui elementi vengono definiti nella routine di ordinamento.

In 730-790 c'è il menù che potremmo definire principale, dal quale si può accedere alle più importanti funzioni svolte dal programma. In 900-990 si definiscono caratteri speciali che verranno utilizzati nella parte grafica; la stampa della tabella dei dati non ordinata occupa le linee da 1020 a 1050; nell'intervallo 1110-1230 si trova invece la routine per l'ordinamento. In essa ogni elemento di K\$ viene confrontato con ogni altro elemento dello stesso vettore; la variabile P è incrementata di una unità ogni volta che l'elemento preso in esame risulta maggiore di un altro ed indica perciò il posto che deve occupare nella lista ordinata. Se, ad esempio, il record 3, in base ai confronti viene a trovarsi al decimo posto, si carica l'elemento X(10) con il valore 3 (linea 1210); in seguito, stampando al decimo posto l'elemento con indice X(10), verrà richiamato proprio il record che in input aveva indice 3.

Tutta la parte compresa fra le linee 1240 e 2070 è in funzione del dimensionamento automatico; la sua relativa complessità è giustificata dal tipo di output che si è voluto ottenere. Se la misura massima introdotta è maggiore di 9999, apparirà la scritta x1000 mentre tutte le misure saranno divise per 1000, oppure verrà stampato x1000000 e tutte le misure saranno ridotte di un milione, e così via. Alla linea 1320, VALUNI contiene il valore unitario di ogni singolo carattere che forma la colonnina nel diagramma; si è assunto che la lunghezza massima di una striscia sia di 18 caratteri. In base al valore unitario (linee 1340-1370) l'asse verticale è suddiviso in 3 o 5 parti corrispondenti a multipli di 1, 2, 3 o 0.5, 1, 1.5 o 1, 2, 3, 4, 5 o ancora, 2, 4, 6, 8, 10.

Sarebbe stato molto più semplice trattare solo multipli di questi ultimi valori, tuttavia, supponendo che il valore massimo da rappresentare sia 140, si sarebbero dovuti adottare i valori 200, 400, 600, 800 e 1000 con un evidente appiattimento di tutto il grafico verso il basso.

Il programma stamperà invece tre suddivisioni con i valori 50, 100, 150 ed il grafico occuperà quasi l'intero schermo permettendo una maggiore definizione dell'immagine.

Nelle linee 2350-2900 si stampano diversi simboli sull'asse orizzontale, mentre la parte seguente è occupata dalla routine per il grafico (2910-3010).

Punti e rette sul piano cartesiano

Il seguente programma fornisce i primi elementi a chi inizia lo studio della geometria analitica ed è già stato utilizzato in classe quale strumento didattico.

Le funzioni svolte sono molteplici: posizioni dei punti date le loro coordinate, con le linee di costruzione tratteggiate; coordinate del punto medio tra due punti dati; distanza tra due punti; grafico di una o due rette simultaneamente, date le loro equazioni esplicite; equazione di una retta passante per due punti; retta di dato coefficiente angolare e passante per un punto; punto di intersezione tra due rette. Vengono individuate condizioni di parallelismo ed è previsto l'input di rette parallele all'asse y , aventi, come è noto, un'equazione esplicita nella forma $x=K$.

I vari elementi sono presentati contemporaneamente in forma grafica ed analitica e vengono contraddistinti da diversi colori.

In accordo con quanto detto all'inizio del capitolo, il programma adotta alcuni accorgimenti volti a migliorare la grafica. Il primo consiste nello stampare un punto per ogni colonna, ma a quattro differenti altezze all'interno di ogni riga in modo da attenuare l'aspetto a "gradini" che assumerebbe una retta inclinata sugli assi.

Il secondo consiste nell'usare la funzione inversa, $x=f(Y)$, quando il valore assunto dal coefficiente angolare della retta è maggiore di 1; questa scelta è giustificata dal fatto che, se si stampa un punto per ogni colonna, per inclinazioni superiori a 45° , i punti che formano la retta sono sempre più distanziati ed è evidente che in questo caso è preferibile stampare un punto per ogni riga.

```

100 REM *****
110 REM *
120 REM * PUNTI E RETTE *
130 REM * SUL PIANO *
140 REM * CARTESIANO. *
150 REM *
160 REM *****
170 REM DI SERGIO BORSANI
180 REM VERSIONE: TI BASIC
190 CALL CLEAR
200 CALL CHAR(128,"181818FFFF181818")
210 CALL CHAR(129,"181818FBFB181818")
220 CALL CHAR(130,"000000FFFF181818")
230 CALL CHAR(131,"0010101010101000")
240 AS="3C18"
250 BS="183C3C18"
260 CS="0000183C3C180000"
270 DS="00000000183C3C18"
280 ES="000000000000183C"
290 FS="000080C0C080"
300 GS="000001030301"
310 HS="0"
320 CALL CHAR(136,AS)
330 CALL CHAR(137,BS)
340 CALL CHAR(138,CS)
350 CALL CHAR(139,DS)
360 CALL CHAR(140,ES)
370 CALL CHAR(141,FS)
380 CALL CHAR(142,GS)
390 CALL CHAR(143,HS)
400 CALL CHAR(144,AS)
410 CALL CHAR(145,BS)
420 CALL CHAR(146,CS)

```

```

430 CALL CHAR(147,D8)
440 CALL CHAR(148,E8)
450 CALL CHAR(149,F8)
460 CALL CHAR(150,G8)
470 CALL CHAR(151,H8)
480 CALL CHAR(152,A8)
490 CALL CHAR(153,B8)
500 CALL CHAR(154,C8)
510 CALL CHAR(155,D8)
520 CALL CHAR(156,E8)
530 CALL CHAR(157,F8)
540 CALL CHAR(158,G8)
550 CALL CHAR(159,H8)
560 CALL COLOR(14,6,1)
570 CALL COLOR(15,7,1)
580 CALL COLOR(16,13,1)
590 GOSUB 3460
600 GOSUB 3110
610 CALL CLEAR
620 CALL COLOR(2,2,1)
630 PRINT "PREMI:"
640 PRINT "1. PUNTI SUL PIANO"
650 PRINT "2. DISTANZA TRA DUE PUNTI"
660 PRINT "3. PUNTO MEDIO"
670 PRINT "4. RETTE DI DATA EQUAZIONE"
680 PRINT "5. EQUAZIONE DI UNA RETTA"
690 PRINT "6. RETTA DI DATO COEFF. AN- PASSANTE PER DUE PUNTI"
700 PRINT "7. PUNTO DI INCONTRO TRA SOLARE PASSANTE PER UN PUNTO "
710 PRINT "8. PER FINIRE":
720 CALL KEY(0,K,STATUS)
730 IF STATUS=0 THEN 720
740 IF K<49 THEN 720
750 IF K>56 THEN 720
760 SCELTA=K-48
770 ON SCELTA GOTO 780,1340,1540,1690,2370,2610,2740,3890
780 CALL CLEAR
790 PRINT "QUANTI PUNTI VUOI SEGNARE, 1,2 O 3?"
800 INPUT N
810 N=INT(N)
820 IF N<1 THEN 850
830 IF N>3 THEN 850
840 GOTO 870
850 PRINT "NON HO CAPITO, PER FAVORE RIPETI":
860 GOTO 800
870 FOR I=1 TO N
880 PRINT "PUNTO"II
890 INPUT " ASCISSA? "P(I,1)
900 INPUT " ORDINATA? "P(I,2)
910 PRINT
920 NEXT I
930 CALL CLEAR
940 GOSUB 3220
950 FOR I=1 TO N
960 FC=0
970 RIGA=10-P(I,2)
980 COL=16-P(I,1)
990 IF RIGA<2 THEN 1170
1000 IF RIGA>18 THEN 1170
1010 IF COL<6 THEN 1170
1020 IF COL>26 THEN 1170
1030 CALL HCHAR(RIGA,COL,130+I*8)
1040 IF COL>17 THEN 1070
1050 IF COL<15 THEN 1070
1060 GOTO 1100
1070 CALL HCHAR(RIGA,17,45,COL-17)
1080 GOTO 1100
1090 CALL HCHAR(RIGA,COL+1,45,15-COL)
1100 IF RIGA>11 THEN 1130
1110 IF RIGA<9 THEN 1150
1120 GOTO 1180
1130 CALL VCHAR(11,COL,131,RIGA-11)
1140 GOTO 1180
1150 CALL VCHAR(RIGA+1,COL,131,9-RIGA)
1160 GOTO 1180
1170 FC=1
1180 PUNTO$="P"&STR$(I)&" "&CHR$(130+I*8)&"("&STR$(P(I,1))&","&STR$(P(I,2))&"
1190 FOR J=1 TO LEN(PUNTO$)
1200 L=ASC(SEG$(PUNTO$,J,1))
1210 CALL HCHAR(19+I,44+J,L)
1220 NEXT J
1230 IF FC<>1 THEN 1300
1240 DATA 70,85,79,82,73,32,67,65,77,80,79,32
1250 RESTORE 1240
1260 FOR J=19 TO 30
1270 READ L
1280 CALL HCHAR(19+I,J,L)

```

```

1290 NEXT J
1300 NEXT I
1310 IF DIST=1 THEN 1440
1320 GOSUB 3110
1330 GOTO 3670
1340 CALL CLEAR
1350 FOR I=1 TO 2
1360 PRINT "COORDINATE PUNTO":I
1370 INPUT "  ASCISSA = ":P(I,1)
1380 INPUT "  ORDINATA = ":P(I,2)
1390 PRINT
1400 NEXT I
1410 N=2
1420 DIST=1
1430 GOTO 930
1440 DIST=0
1450 DI2=(P(1,1)-P(2,1))^2+(P(1,2)-P(2,2))^2
1460 DI=INT(1000*SDR(DI2))/1000
1470 DIS$="DISTANZA = "&STR$(DI)
1480 FOR I=1 TO LEN(DIS$)
1490 L=ASC(SEG$(DIS$,I,1))
1500 CALL MCHAR(22,4+I,L)
1510 NEXT I
1520 GOSUB 3110
1530 GOTO 3670
1540 CALL CLEAR
1550 FOR I=1 TO 2
1560 PRINT "COORDINATE PUNTO":I
1570 INPUT "  ASCISSA = ":P(I,1)
1580 INPUT "  ORDINATA = ":P(I,2)
1590 PRINT
1600 NEXT I
1610 PRINT "XM=(X1+X2)/2 ; YM=(Y1+Y2)/2"
1620 PRINT "::"
1630 XM=(P(1,1)+P(2,1))/2
1640 YM=(P(1,2)+P(2,2))/2
1650 PRINT "COORDINATE PUNTO MEDIO:"
1660 PRINT "M(";INT(100*XM)/100;",";INT(100*YM)/100;")::"
1670 GOSUB 3110
1680 GOTO 3670
1690 CALL CLEAR
1700 PRINT "EQUAZIONE ESPLICITA DELLA RETTA::"
1710 PRINT "Y=MX+N::"
1720 PRINT "M=COEFFICIENTE ANGOLARE"
1730 PRINT "N=ORDINATA ALL'ORIGINE::"
1740 PRINT "VUOI OTTENERE 1 O 2 RETTE?"
1750 INPUT N
1760 IF N=1 THEN 1800
1770 IF N=2 THEN 1800
1780 PRINT "SPIACENTE, NON POSSO FARLO"
1790 GOTO 1750
1800 FOR I=1 TO N
1810 PRINT "PARAMETRI DELLA RETTA":I
1820 INPUT "  LA RETTA E' PARALLELA"
1830 IF SEG$(RISP$,I,1)<>"S" THEN 1860
1840 GOSUB 3720
1850 GOTO 1880
1860 INPUT "  M = ":M(I)
1870 INPUT "  N = ":N(I)
1880 PRINT
1890 NEXT I
1900 CALL CLEAR
1910 GOSUB 3220
1920 FOR I=1 TO N
1930 IF PA(I)<>1 THEN 1960
1940 GOSUB 3770
1950 GOTO 2330
1960 IF M(I)>1 THEN 2160
1970 IF M(I)<-1 THEN 2160
1980 FOR COL=6 TO 26
1990 RIBA=10-(M(I)*(COL-16)+N(I))
2000 IF RIBA<2 THEN 2140
2010 IF RIBA>18 THEN 2140
2020 DIFF=RIBA-INT(RIBA)
2030 IF DIFF>.7 THEN 2080
2040 IF DIFF>.45 THEN 2100
2050 IF DIFF>.2 THEN 2130
2060 CALL MCHAR(INT(RIBA),COL,130+I*8)
2070 GOTO 2140
2080 CALL MCHAR(INT(RIBA)+1,COL,129+I*8)
2090 GOTO 2140
2100 CALL MCHAR(INT(RIBA)+1,COL,128+I*8)
2110 CALL MCHAR(INT(RIBA),COL,132+I*8)
2120 GOTO 2140
2130 CALL MCHAR(INT(RIBA),COL,131+I*8)
2140 NEXT COL

```

```

2150 GOTO 2270
2160 FOR RIGA=2 TO 18
2170 COL=16+(10-RIGA-Q(1))/M(1)
2180 IF COL<6 THEN 2260
2190 IF COL>26 THEN 2260
2200 DIFF=COL-INT(COL)
2210 IF DIFF>.499 THEN 2240
2220 CALL HCHAR(RIGA,INT(COL),130+I*B)
2230 GOTO 2260
2240 CALL HCHAR(RIGA,INT(COL),134+I*B)
2250 CALL HCHAR(RIGA,INT(COL)+1,133+I*B)
2260 NEXT RIGA
2270 RET$="RETTA "&STR$(I)&" Y=("&STR$(M(1))&")*X+("&STR$(Q(1))&")"
2280 FOR K=1 TO LEN(RET$)
2290 L=ASC(SEG$(RET$,K,1))
2300 CALL HCHAR(19+I,3+K,L)
2310 NEXT K
2320 GOTO 2340
2330 GOSUB 3B20
2340 NEXT I
2350 GOSUB 3110
2360 GOTO 3670
2370 CALL CLEAR
2380 PRINT "RETTA PASSANTE PER DUE PUNTI"!!!
2390 PRINT "SCRIVI LE COORDINATE DEI DUEPUNTI PER I QUALI PASSA LA RETTA"!!
2400 INPUT "X1,Y1 = ":X1,Y1
2410 INPUT "X2,Y2 = ":X2,Y2
2420 IF X1<>X2 THEN 2510
2430 PRINT "EQUAZIONE DELLA RETTA:"
2440 PRINT "Y="&STR$(X1)!!
2450 INPUT "VUOI IL GRAFICO DELLA RETTA?":RI$
2460 IF SEG$(RI$,1,1)<>"S" THEN 610
2470 N=1
2480 PA(1)=1
2490 W(1)=X1
2500 GOTO 1900
2510 MM=(Y2-Y1)/(X2-X1)
2520 NN=(Y1-X2-X1*Y2)/(X2-X1)
2530 PRINT "EQUAZIONE DELLA RETTA:"!!
2540 PRINT "Y="&STR$(INT(100*MM)/100)!!)*X+("&STR$(INT(100*NN)/100)!!)"!!
2550 INPUT "VUOI IL GRAFICO DELLA RETTA?":RI$
2560 IF SEG$(RI$,1,1)<>"S" THEN 610
2570 N=1
2580 M(1)=INT(100*MM)/100
2590 Q(1)=INT(100*NN)/100
2600 GOTO 1900
2610 CALL CLEAR
2620 PRINT "RETTA DI DATO COEFF. ANGOLA-RE E PASSANTE PER UN PUNTO."!!!
2630 PRINT "SCRIVI IL COEFF. ANGOLARE IN FORMA DECIMALE"
2640 INPUT "M = ":M(1)
2650 PRINT "COORDINATE DEL PUNTO"
2660 INPUT "X,Y=":X1,Y1
2670 D(1)=Y1-M(1)*X1
2680 PRINT "EQUAZIONE DELLA RETTA:"
2690 PRINT "Y=("&STR$(M(1))!!)*X+("&STR$(INT(Q(1)))!!)"!!
2700 INPUT "VUOI IL GRAFICO DELLA RETTA?":RI$
2710 IF SEG$(RI$,1,1)<>"S" THEN 610
2720 N=1
2730 GOTO 1900
2740 CALL CLEAR
2750 PAR(1)=0
2760 PAR(2)=0
2770 PRINT "PUNTO DI INTERSEZIONE TRA DUE RETTE":::
2780 PRINT "Y=M*X+N"!!
2790 FOR I=1 TO 2
2800 PRINT "PARAMETRI DELLA RETTA"!!
2810 INPUT "LA RETTA E' PARALLELA ALL'ASSE Y?":RI$
2820 IF SEG$(RI$,1,1)<>"S" THEN 2860
2830 GOSUB 3720
2840 PAR(I)=1
2850 GOTO 2880
2860 INPUT "M = ":M(1)
2870 INPUT "N = ":Q(1)
2880 PRINT
2890 NEXT I
2900 SOMMA=PAR(1)+PAR(2)+1
2910 ON SOMMA GOTO 2920,3030,3070,2930
2920 IF M(1)<>M(2) THEN 2950
2930 PRINT "LE RETTE SONO PARALLELE":::
2940 GOTO 2990
2950 X=(Q(2)-Q(1))/(M(1)-M(2))
2960 Y=M(1)*X+Q(1)
2970 PRINT "COORDINATE DEL PUNTO INTERSEZIONE:"!!
2980 PRINT "("&STR$(INT(100*X)/100)!!","&STR$(INT(100*Y)/100)!!)"!!
2990 INPUT "VUOI IL GRAFICO DELLE RETTE?":RI$
3000 IF SEG$(RI$,1,1)<>"S" THEN 610
3010 N=2

```

```

3020 GOTO 1900
3030 Y=M(2)*M(1)+Q(2)
3040 PRINT "COORDINATE DEL PUNTO INTERSEZIONE:";
3050 PRINT " (";INT(100*M(1)/100);", ";INT(100*Y/100);" )";
3060 GOTO 2990
3070 Y=M(1)*M(2)+Q(1)
3080 PRINT "COORDINATE DEL PUNTO INTERSEZIONE:";
3090 PRINT " (";INT(100*M(2)/100);", ";INT(100*Y/100);" )";
3100 GOTO 2990
3110 CALL SOUND(225,784,2)
3120 CALL SOUND(150,659,2)
3130 CALL SOUND(300,740,2)
3140 CALL SOUND(300,587,2)
3150 CALL SOUND(300,494,2)
3160 CALL SOUND(300,784,2)
3170 CALL SOUND(1000,659,2)
3180 CALL SOUND(300,587,2)
3190 CALL SOUND(600,587,2)
3200 CALL SOUND(300,587,2)
3210 RETURN
3220 REM ***ASSI CARTESIANI***
3230 CALL HCHAR(10,6,130,10)
3240 CALL HCHAR(10,16,128)
3250 CALL HCHAR(10,17,130,10)
3260 CALL VCHAR(2,16,129,8)
3270 CALL VCHAR(11,16,129,8)
3280 CALL HCHAR(9,26,88)
3290 CALL HCHAR(2,17,89)
3300 CALL HCHAR(11,5,45)
3310 CALL HCHAR(11,6,49)
3320 CALL HCHAR(11,7,48)
3330 CALL HCHAR(11,10,45)
3340 CALL HCHAR(11,11,53)
3350 CALL HCHAR(11,15,48)
3360 CALL HCHAR(11,20,43)
3370 CALL HCHAR(11,21,53)
3380 CALL HCHAR(11,25,43)
3390 CALL HCHAR(11,26,49)
3400 CALL HCHAR(11,27,48)
3410 CALL HCHAR(15,14,45)
3420 CALL HCHAR(15,15,53)
3430 CALL HCHAR(5,14,43)
3440 CALL HCHAR(5,15,53)
3450 RETURN
3460 REM ***SUB. TITOLO***
3470 T1$="PUNTI E RETTE"
3480 T2$="GEOMETRIA ANALITICA"
3490 FOR J=1 TO LEN(T2$)
3500 L=ASC(SEG$(T2$,J,1))
3510 CALL HCHAR(16,6+J,L)
3520 NEXT J
3530 FOR J=1 TO LEN(T1$)
3540 L=ASC(SEG$(T1$,J,1))
3550 CALL HCHAR(16,9+J,L)
3560 NEXT J
3570 CALL COLOR(2,16,5)
3580 CALL HCHAR(14,5,42,23)
3590 CALL VCHAR(15,27,42,6)
3600 CALL VCHAR(15,5,42,6)
3610 CALL HCHAR(20,5,42,23)
3620 FOR K=1 TO 20
3630 CALL COLOR(2,5,16)
3640 CALL COLOR(2,16,5)
3650 NEXT K
3660 RETURN
3670 PRINT " * PREMI I. PER PROSEGUIRE "
3680 CALL KEY(0,K,STATUS)
3690 IF STATUS=0 THEN 3680
3700 CALL CLEAR
3710 GOTO 630
3720 REM ***RETTA PARALL. ASSE Y***
3730 PRINT "X=K, K=DISTANZA DALL'ASSE Y";
3740 INPUT K="M(I)"
3750 PA(I)=;
3760 RETURN
3770 REM ***GRAFICO RETTA PARALL. ASSE Y***
3780 COL=16+M(I)
3790 CALL VCHAR(2,COL,130+I*8,17)
3800 PA(I)=0
3810 RETURN
3820 REM ***EQUAZ. RETTA PARALL. ASSE Y***
3830 RET$="RETTA "&STR$(I)&": X="&STR$(M(I))
3840 FOR K=1 TO LEN(RET$)
3850 L=ASC(SEG$(RET$,K,1))
3860 CALL HCHAR(19+I,3+K,L)
3870 NEXT K

```

```

0880 RETURN
0890 CALL CLEAR
0900 PRINT " F I N E "
0910 GOSUB 3110
0920 END

```

Il programma ha la seguente struttura: definizione caratteri speciali - menù - varie sezioni con funzioni diverse - subroutine. Agli stessi caratteri sono stati attribuiti più numeri appartenenti a diversi sottoinsiemi in modo da poterli rappresentare in colori diversi. Due rette presenti contemporaneamente sullo schermo saranno date, quindi, con differenti colori e questi serviranno da riferimento per associare loro le equazioni esplicite. Alle linee 630-710 si trova il menù principale. La scelta viene accolta con l'istruzione CALL KEY, pertanto non è necessario premere il tasto ENTER; alla linea 770 un *branch* rinvia alle varie sezioni. La prima di queste occupa le linee da 780 a 1330 ed in essa viene dato il concetto di coordinate di un punto. L'origine degli assi cartesiani è posta al centro dello schermo per visualizzare i quattro quadranti e permettere l'uso di coordinate con valore negativo. Sul piano cartesiano trovano posto fino a tre punti contemporaneamente. Da linea 1070 a 1150 vengono stampate linee di costruzione tratteggiate, parallele agli assi. In 1170 il flag FC viene posto uguale a 1 se il punto è caduto fuori campo e quindi non è stato stampato. Nelle linee 1180-1220 viene dato il colore di ogni punto e, a fianco, le sue coordinate; se FC=1 appare la scritta FUORI CAMPO. Per queste operazioni non è stata usata l'istruzione PRINT poiché, in tal modo, il grafico si sarebbe spostato verso l'alto.

La seconda sezione occupa le righe da 1340 a 1530, in essa si trova la distanza tra due punti.

La distanza si trova applicando il teorema di Pitagora (linea 1450) e il valore viene dato con non più di tre cifre dopo la virgola (linea 1470). Nelle linee 1540-1680 si trova il punto medio tra due punti dati. XM e YM (1630-1640) contengono i valori medi rispettivamente delle ascisse e delle ordinate, che costituiscono, appunto, le coordinate del punto medio.

Da 1690 a 2360 trova posto la prima sezione dedicata alle rette. Il valore di N (INPUT di linea 1750) indica se si vuole una sola retta o due contemporaneamente. I coefficienti angolari M(I) e le ordinate all'origine Q(I) devono essere introdotti in forma decimale, non sotto forma di frazione. Il valore di PA(I) è 1 se il programma passa alla subroutine di linea 3720, nell'eventualità che una retta sia parallela all'asse delle Y. In 1960, se il coefficiente angolare M(I) è maggiore di 1 il programma considera la funzione inversa e si pone un punto in ogni riga piuttosto che in ogni colonna.

Nelle righe 2020 e seguenti, in base al valore della variabile DIFF, si stampa il carattere più idoneo a rappresentare la retta in quel punto.

Da 2370 a 2600, accettate in input le coordinate dei due punti, viene da-

ta l'equazione della retta passante per essi (2540) e, su richiesta, la sua rappresentazione grafica. In 2420 si distingue se i due punti hanno la stessa ascissa, poiché in tal caso la retta risulta parallela all'asse Y. Da 2610 a 2730 trova posto la breve sezione per trovare l'espressione analitica di una retta di un dato coefficiente angolare passante per un punto; l'ordinata all'origine è calcolata alla linea 2670; per ottenere il grafico si rimanda sempre alla linea 1900. Nell'intervallo 2740-3880 si calcola il punto di intersezione tra due rette. Prima di introdurre i valori dei parametri viene chiesto se la retta è parallela all'asse Y poiché, in tal caso, si passa alla routine di riga 3720. La variabile PAR(1) memorizza quali rette siano parallele all'asse Y, inoltre, la condizione di parallelismo tra le due rette è rilevata dal confronto dei loro coefficienti angolari (linea 2920). Se le rette si intersecano, le coordinate del punto di incontro vengono calcolate alle linee 2950-2960. Seguono le linee dedicate all'output. Le subroutine occupano l'ultima parte del programma. Alla riga 3110 un breve commento musicale; non è essenziale, ma occupa una minima parte di memoria. A 3220 c'è la routine per tracciare gli assi cartesiani; i caratteri speciali sono stati definiti all'inizio del programma. Alla linea 3460 viene scritto il titolo; anche qui si è fatta qualche concessione all'estetica, ma il TI 99/4A, con i suoi 16 K, non presenta assillanti problemi di memoria. Alla linea 3670, una routine per stampare PREMI 1 PER PROSEGUIRE, un messaggio ricorrente al termine di varie sezioni. Alle linee 3720, 3770 e 3820 tre brevi routine che vengono richiamate quando le rette sono parallele all'asse Y; in questo caso, come è già stato detto, l'equazione è del tipo $x=K$ e la stampa dei punti che costituiscono il grafico può essere fatta con l'istruzione CALL VCHAR (3790). Nelle ultime righe si stampa la parola FINE e termina il programma.

Parabole

Mentre il precedente programma era prevalentemente indirizzato ad uno studente di scuola media inferiore, quello che segue è più adatto a chi segue studi superiori.

La parabola è il luogo geometrico dei punti equidistanti da un punto fisso, detto fuoco, e da una retta, detta direttrice. Analiticamente è data dall'equazione di secondo grado:

$$y = ax^2 + bx + c$$

Le coordinate del vertice sono legate alle costanti a , b , c dalle seguenti uguaglianze:

$$x = -\frac{b}{2a} \quad y = -\frac{b^2 - 4ac}{4a}$$

Il fuoco ha la stessa ascissa del vertice e ordinata

$$y = \frac{1}{4a} - \frac{b^2 - 4ac}{4a}$$

La direttrice è una retta di equazione:

$$y = -\frac{1}{4a} - \frac{b^2 - 4ac}{4a}$$

L'asse di simmetria è una retta di equazione:

$$x = -\frac{b}{2a}$$

Il programma, accettati in input i valori delle costanti a , b , c , fornisce tutti gli elementi su esposti e i punti di intersezione con gli assi cartesiani. La costante c fornisce il punto di intersezione con l'asse y , mentre le intersezioni con l'asse x sono date, se esistono, dalle soluzioni reali dell'equazione di secondo grado

$$ax^2 + bx + c = 0$$

Il programma fornisce inoltre i punti di intersezione tra una parabola ed una retta generica, anche parallela all'asse y . Le coordinate di tali punti sono le soluzioni del sistema tra l'equazione della parabola e quella della retta.

Un'ulteriore funzione svolta dal programma è lo studio analitico di una parabola passante per tre punti dati. A tale riguardo si noti che se una parabola passa per il punto di coordinate (3,2), sostituendo questi valori nell'equazione generica della parabola, si ottiene:

$$2 = 9a + 3b + c$$

Quindi, dati tre punti, si ottengono 3 equazioni di primo grado a tre incognite che, poste a sistema, danno i valori di a , b , c .

Per quanto riguarda la grafica, è stato adottato l'accorgimento descritto all'inizio del capitolo, con il risultato di avere l'output su una "finestra" dotata di 56 colonne e 80 righe. Inoltre è previsto un cambiamento di scala che permette un dimensionamento grafico definibile dall'utente; questo, rispetto al dimensionamento automatico, presenta il vantaggio di consentire una visione d'insieme o di una parte, ingrandita, del grafico. Un'ultima particolarità: l'equazione della retta definita nel programma ma si ottiene di volta in volta ponendo uguale a zero il coefficiente del termine di secondo grado nell'equazione della parabola.

```

100 REM *****
110 REM *
120 REM *   GEOMETRIA *
130 REM *   ANALITICA *
140 REM *   - *
150 REM *   PARABOLE. *
160 REM *
170 REM *****
180 REM di: SERGIO BORSANI
190 REM versione: TI BASIC
200 CALL CLEAR
210 CALL CHAR(128,"0000000000000000")
220 CALL CHAR(129,"0000000000000000")
230 CALL CHAR(130,"00000000")
240 CALL CHAR(131,"0000")
250 CALL CHAR(132,"0000000000000000")
260 CALL CHAR(133,"0000000000000000")
270 CALL CHAR(134,"0000000000000000")
280 CALL CHAR(135,"0000000000000000")
290 CALL CHAR(136,"0000000000000000")
300 CALL CHAR(137,"000000000000")
310 CALL CHAR(138,"0000000000000000")
320 CALL CHAR(139,"000000000000")
330 CALL CHAR(140,"000000000000")
340 CALL CHAR(141,"000000000000")
350 CALL CHAR(142,"00000000")
360 CALL CHAR(143,"0000000000000000")
370 CALL CHAR(144,"000000000000")
380 CALL CHAR(145,"00000000")
390 CALL CHAR(146,"00000000")
400 CALL CHAR(147,"0000")
410 CALL CHAR(148,"0000000000000000")
420 CALL CHAR(149,"000000000000")
430 CALL CHAR(150,"00000000")
440 CALL CHAR(151,"0000")
450 CALL CHAR(152,"0101010101010101")
460 CALL CHAR(153,"FF01")
470 CALL CHAR(154,"0F01010101010101")
480 CALL CHAR(155,"FF010101")
490 CALL CHAR(156,"FF01010101010101")
500 CALL CHAR(157,"0")
510 CALL CHAR(158,"0000000000000000")
520 CALL CHAR(159,"0000000000000000")
530 CALL CHAR(160,"00000000")
540 CALL CHAR(161,"000000000000")
550 CALL CHAR(162,"000000000000")
560 DEF Y=A*X+X*B+C
570 REM * TITOLO *
580 FOR C=12 TO 15
590 CALL COLOR(C,2,16)
600 NEXT C
610 CALL COLOR(16,9,16)
620 FOR RIGA=4 TO 22
630 CALL HCHAR(RIGA,3,157,12)
640 NEXT RIGA
650 CALL HCHAR(13,3,153,12)
660 CALL VCHAR(4,8,152,19)
670 CALL HCHAR(13,8,156)
680 A=-.7
690 B=-10
700 C=-32
710 GOSUB 2430
720 A$="GEOMETRIA"
730 FOR J=1 TO LEN(A$)
740 L=ASC(SEG$(A$,J,1))
750 CALL HCHAR(12,16+J,L)
760 NEXT J
770 B$="ANALITICA"
780 FOR J=1 TO LEN(B$)
790 L=ASC(SEG$(B$,J,1))
800 CALL HCHAR(14,16+J,L)
810 NEXT J
820 C$="PARABOLE"
830 FOR J=1 TO LEN(C$)
840 L=ASC(SEG$(C$,J,1))
850 CALL HCHAR(18,16+J,L)
860 NEXT J
870 CALL SOUND(150,659,2)
880 CALL SOUND(150,587,2)
890 CALL SOUND(300,523,2)
900 CALL SOUND(300,440,2)
910 CALL SOUND(300,698,2,440,5)
920 CALL SOUND(300,784,2,587,5)
930 CALL SOUND(150,698,2,392,5)
940 CALL SOUND(150,659,2)
950 CALL SOUND(1000,698,2,440,5)

```

```

960 FOR T=1 TO 1000
970 NEXT T
980 CALL CLEAR
990 PRINT "PREMI:":
1000 PRINT "1 STUDIO DELLA PARABOLA":
1010 PRINT "2 INTERSEZIONE CON UNA RETTA":
1020 PRINT "3 PARABOLA PASSANTE PER TRE PUNTI":
1030 PRINT "4 FINE":
1040 CALL KEY(0,K,S)
1050 IF S=0 THEN 1040
1060 IF K=49 THEN 1110
1070 IF K=50 THEN 1680
1080 IF K=51 THEN 2970
1090 IF K=52 THEN 4490
1100 GOTO 1040
1110 REM *STUDIO PARABOLA*
1120 CALL CLEAR
1130 SCLTA=1
1140 PRINT "EQUAZIONE DELLA PARABOLA":
1150 PRINT TAB(9); "Y=A*X^2+B*X+C":
1160 CALL MCHAR(24,16,35)
1170 PRINT TAB(15); "+B*X+C":
1180 GOSUB 2710
1190 CALL CLEAR
1200 VX=INT((-B+1000/(2*A))/1000)
1210 DELTA=B^2-4*A*C
1220 VY=INT((-DELTA+1000/(4*A))/1000)
1230 PRINT "COORDINATE DEL VERTICE":
1240 PRINT "V = ("XV","YV")":
1250 PRINT "ASSE DI SIMEITIA":
1260 PRINT "X = "XV":
1270 YF=INT(1000*(1/(4*A)-DELTA/(4*A))/1000)
1280 PRINT "COORDINATE DEL FUOCO":
1290 PRINT "F = ("XF","YF")":
1300 YD=INT(1000*(-1/(4*A)-DELTA/(4*A))/1000)
1310 PRINT "DIRETTRICE":
1320 PRINT "Y = "YD":
1330 GOSUB 4010
1340 PRINT "INTERSEZIONE CON L'ASSE Y":
1350 PRINT "Y = "C":
1360 IF DELTA>0 THEN 1390
1370 PRINT "LA PARABOLA NON INTERSECA L'ASSE DELLE X":
1380 GOTO 1430
1390 X1=(-B-SQR(DELTA))/(2*A)
1400 X2=(-B+SQR(DELTA))/(2*A)
1410 PRINT "INTERSEZIONE CON L'ASSE X":
1420 PRINT "X1 = "INT(1000*X1)/1000,"X2 = "INT(1000*X2)/1000:
1430 INPUT "VUOI IL GRAFICO? ":R#
1440 IF SEG$(R#,1,1)="S" THEN 1460
1450 GOTO 980
1460 SCALA=1
1470 GOSUB 4290
1480 GOSUB 2430
1490 IF SCLTA=2 THEN 2170
1500 IF SCLTA=3 THEN 3790
1510 GOSUB 4110
1520 GOSUB 4010
1530 INPUT "VUOI IL GRAFICO CON UN ALTROFATTORE DI SCALA? (S/N) ":R#
1540 IF SEG$(R#,1,1)="S" THEN 1560
1550 GOTO 980
1560 PRINT :
1570 A=A/SCALA
1580 C=C/SCALA
1590 KK=KK*SCALA
1600 INPUT "SCALA? (N>0) ":SCALA
1610 IF SCALA<=0 THEN 1600
1620 A=A*SCALA
1630 C=C/SCALA
1640 IF SCLTA=3 THEN 3950
1650 CALL CLEAR
1660 GOTO 1470
1670 REM *INTERSEZIONE*
1680 CALL CLEAR
1690 PRINT "INTERSEZIONE CON UNA RETTA":
1700 VERT=0
1710 INPUT "LA RETTA E' PARALLELA ALL'ASSE DELLE Y? (S/N) ":R#
1720 IF SEG$(R#,1,1)="N" THEN 1740
1730 VERT=1
1740 PRINT :
1750 INPUT "VUOI SCRIVERE I COEFFICIENTI IN FORMA FRAZIONARIA? ":R#
1760 IF SEG$(R#,1,1)="N" THEN 1930
1770 PRINT "PARABOLA":
1780 GOSUB 2770
1790 PRINT :
1800 IF VERT=1 THEN 3620
1810 PRINT "RETTA":

```

```

1820 INPUT "COEFF. ANGOLARE" (M1,M2) ":M1,M2
1830 IF M2<>0 THEN 1860
1840 PRINT "ERRORE!":
1850 GOTO 1820
1860 M=M1/M2
1870 INPUT "TERMINE NOTO" (Q1,Q2) ":Q1,Q2
1880 IF Q2<>0 THEN 1910
1890 PRINT "ERRORE!":
1900 GOTO 1870
1910 Q=Q1/Q2
1920 GOTO 1990
1930 PRINT "PARABOLA":
1940 INPUT "(A,B,C) ":A,B,C
1950 PRINT ":
1960 IF VERT=1 THEN 3700
1970 PRINT "RETTA":
1980 INPUT "(M,Q) ":M,Q
1990 PRINT ":
2000 A0=A
2010 B0=B
2020 C0=C
2030 DISCR=(B-M)^2-4*A*(C-Q)
2040 IF DISCR<0 THEN 2080
2050 PRINT "NON ESISTONO PUNTI DI INTER-SEZIONE":
2060 GOSUB 4010
2070 GOTO 2150
2080 X1=(-(B-M)-SOR(DISCR))/(2*A)
2090 X2=(-(B-M)+SOR(DISCR))/(2*A)
2100 Y1=M*X1+Q
2110 Y2=M*X2+Q
2120 PRINT "PUNTI DI INTERSEZIONE":
2130 PRINT "P1 = (":INT(1000*X1)/1000;"",INT(1000*Y1)/1000:")":
2140 PRINT "P2 = (":INT(1000*X2)/1000;"",INT(1000*Y2)/1000:")":
2150 SCALA=2
2160 GOTO 1430
2170 A=0
2180 B=0
2190 C=0
2200 GOSUB 2470
2210 GOSUB 4190
2220 GOSUB 4010
2230 INPUT "VUOI IL GRAFICO CON UN ALTRO FATTORE DI SCALA? (S/N) ":P#
2240 IF SEG$(RS,1,1)="S" THEN 2260
2250 GOTO 980
2260 PRINT ":
2270 AA=AA/SCALA
2280 CC=CC/SCALA
2290 Q=Q/SCALA
2300 INPUT "SCALA? (N>0) ":SCALA
2310 IF SCALA<=0 THEN 2300
2320 AA=AA*SCALA
2330 CC=CC*SCALA
2340 Q=Q*SCALA
2350 A=AA
2360 B=BB
2370 C=CC
2380 CALL CLEAR
2390 GOSUB 4290
2400 GOSUB 2430
2410 GOTO 2170
2420 REM "ROUTINE STAMPA"
2430 FOR XX=-12 TO 12
2440 X=XX-.75
2450 IF ABS(Y)>10 THEN 2550
2460 DIFF=Y-INT(Y)
2470 IF DIFF<.1 THEN 2510
2480 RIGA1=INT(Y)+1
2490 P1=INT(DIFF*4)+1
2500 GOTO 2530
2510 RIGA1=INT(Y)
2520 P1=4
2530 CALL HCHAR(13-RIGA1,16+XX,127+P1)
2540 GOTO 2560
2550 P1=0
2560 Y=XX-.25
2570 IF ABS(Y)>10 THEN 2680
2580 DIFF=Y-INT(Y)
2590 IF DIFF<.1 THEN 2630
2600 RIGA2=INT(Y)+1
2610 P2=(INT(DIFF*4)+1)*5
2620 GOTO 2650
2630 RIGA2=INT(Y)
2640 P2=20
2650 IF RIGA1=RIGA2 THEN 2670
2660 P1=0
2670 CALL HCHAR(13-RIGA2,16+XX,127+P1+P2)

```

```

2680 NEXT XX
2690 RETURN
2700 REM * INPUT A,B,C *
2710 INPUT "VUOI SCRIVERE LE COSTANTI INFORMA FRAZIONARIA? (S/N) ":R$
2720 PRINT "":
2730 IF SEG$(R$,1,1)="S" THEN 2760
2740 INPUT "COSTANTI? (A,B,C) ":A,B,C
2750 GOTO 2920
2760 REM *FRAZIONI*
2770 INPUT " A = (A1,A2) ":A1,A2
2780 IF A2<>0 THEN 2890
2790 PRINT "ERRORE!!!":::
2800 GOTO 2770
2810 INPUT " B = (B1,B2) ":B1,B2
2820 IF B2<>0 THEN 2890
2830 PRINT "ERRORE!!!":::
2840 GOTO 2810
2850 INPUT " C = (C1,C2) ":C1,C2
2860 IF C2<>0 THEN 2890
2870 PRINT "ERRORE!!!":::
2880 GOTO 2850
2890 A=A1/A2
2900 B=B1/B2
2910 C=C1/C2
2920 IF A<>0 THEN 2950
2930 PRINT "NON RISULTA UNA PARABOLA!":
2940 GOTO 2710
2950 RETURN
2960 REM *PARABOLA PER 3 P*
2970 CALL CLEAR
2980 SCLTA=3
2990 PRINT "PARABOLA PASSANTE PER TRE PUNTI.":
3000 INPUT "VUOI SCRIVERE LE COORDINATE IN FORMA FRAZIONARIA? ":R$
3010 IF SEG$(R$,1,1)="S" THEN 3070
3020 PRINT "":
3030 PRINT "PUNTO 1:":
3040 INPUT "COORDINATE? (X,Y) ":X1,Y1
3050 PRINT "PUNTO 2:":
3060 INPUT "COORDINATE? (X,Y) ":X2,Y2
3070 PRINT "PUNTO 3:":
3080 INPUT "COORDINATE? (X,Y) ":X3,Y3
3090 PRINT "":
3100 D1=X1*X1
3110 D2=X1
3120 D3=1
3130 D4=X2*X2
3140 D5=X2
3150 D6=1
3160 D7=X3*X3
3170 D8=X3
3180 D9=1
3190 GOSUB 3990
3200 DD=DETERM
3210 IF DD<>0 THEN 3270
3220 PRINT "":
3230 PRINT "SPIACENTE, NON POSSO DARE LARISPOSTA.":
3240 FOR T=1 TO 1000
3250 NEXT T
3260 GOTO 980
3270 D1=Y1
3280 D4=Y2
3290 D7=Y3
3300 GOSUB 3990
3310 DD1=DETERM
3320 D1=X1*X1
3330 D4=X2*X2
3340 D7=X3*X3
3350 D2=Y1
3360 D5=Y2
3370 D8=Y3
3380 GOSUB 3990
3390 DD2=DETERM
3400 D2=X1
3410 D5=X2
3420 D8=X3
3430 D3=Y1
3440 D6=Y2
3450 D9=Y3
3460 GOSUB 3990
3470 DD3=DETERM
3480 A=INT(1000*(DD1/DD))/1000
3490 B=INT(1000*(DD2/DD))/1000
3500 C=INT(1000*(DD3/DD))/1000
3510 PRINT " Y=A*X ";
3520 CALL HCHAR(24,14,35)
3530 PRINT TAB(13);"+B*X+C":

```

```

3540 PRINT "A =" ; A
3550 PRINT "B =" ; B
3560 PRINT "C =" ; C : : : :
3570 SCELTA=4
3580 INPUT "VUOI PROSEGUIRE CON LO STUDIO DELLA PARABOLA? " : R$
3590 IF SEG$(R$,1,1) = "S" THEN 1190
3600 GOTO 1430
3610 REM "RETTA VERTICALE"
3620 PRINT "RETTA:" : :
3630 PRINT TAB(10); "X=" : : :
3640 INPUT "SCRIVI K IN FORMA FRAZIONARIA (KN,KD) " : KN,KD
3650 IF KD > 0 THEN 3680
3660 PRINT : "ERRORE!!!" : :
3670 GOTO 3640
3680 KK=KN/KD
3690 GOTO 3730
3700 PRINT "RETTA:" : :
3710 PRINT TAB(10); "X=" : : :
3720 INPUT "K = " : KK
3730 PRINT : :
3740 K=KK
3750 PRINT "PUNTO DI INTERSEZIONE:" : :
3760 PRINT "P = (" ; INT(1000*KK)/1000 ; " , " ; INT(1000*Y)/1000 ; " )" : : :
3770 SCELTA=3
3780 GOTO 1430
3790 DF=KK-INT(KK)
3800 IF DF > .1 THEN 3830
3810 CLN=16+KK
3820 GOTO 3920
3830 CLN=17+KK
3840 DDF=INT(DF*4)+1
3850 ON DDF GOTO 3860,3880,3900,3920
3860 NC=158
3870 GOTO 3930
3880 NC=159
3890 GOTO 3930
3900 NC=16
3910 GOTO 3930
3920 NC=17
3930 CALL VCHAR(3,CLN,NC,20)
3940 GOTO 1510
3950 KL=K/SCALA
3960 CALL CLEAR
3970 GOTO 1470
3980 REM "DETERMINANTE"
3990 DETERM=D1*(D5*D9-D8*D6)-D2*(D4*D9-D7*D6)+D3*(D4*D8-D7*D5)
4000 RETURN
4010 CC="" : FREMI 1 PER CONTINUARE"
4020 FOR J=1 TO LEN(CC)
4030 L=ASC(SEG$(CC,J,1))
4040 CALL HCHAR(24,3+J,L)
4050 NEXT J
4060 CALL KEY(0,K,B)
4070 IF S=0 THEN 4060
4080 IF K < > 49 THEN 4060
4090 CALL CLEAR
4100 RETURN
4110 CALL SOUND(150,440,2)
4120 CALL SOUND(150,466,2)
4130 CALL SOUND(300,523,2)
4140 CALL SOUND(300,440,2)
4150 CALL SOUND(300,587,2)
4160 CALL SOUND(300,784,2)
4170 CALL SOUND(500,659,2)
4180 RETURN
4190 CALL SOUND(300,698,2)
4200 CALL SOUND(300,932,2)
4210 CALL SOUND(300,784,2)
4220 CALL SOUND(300,880,2)
4230 CALL SOUND(150,932,2)
4240 CALL SOUND(150,880,2)
4250 CALL SOUND(150,784,2)
4260 CALL SOUND(150,880,2)
4270 CALL SOUND(500,698,2)
4280 RETURN
4290 REM "ASSI CARTESIANI"
4300 CALL CLEAR
4310 FOR CO=13 TO 15
4320 CALL COLOR(CO,2,1)
4330 NEXT CO
4340 CALL COLOR(16,9,1)
4350 CALL HCHAR(13,3,153,28)
4360 CALL VCHAR(3,16,152,20)
4370 FOR COLON=6 TO 26 STEP 5
4380 CALL HCHAR(13,COLON,155)
4390 NEXT COLON

```

```

4400 CALL HCHAR(13,16,156)
4410 CALL HCHAR(8,16,154)
4420 CALL HCHAR(18,16,154)
4430 SCALE="SCALA= 1 : "&STR$(SCALE)
4440 FOR J=1 TO LEN(SCALE)
4450 L=ASC(SEG$(SCALE,J,1))
4460 CALL HCHAR(1,2+J,L)
4470 NEXT J
4480 RETURN
4490 CALL CLEAR
4500 PRINT "          FINE" : GOTO 4550
4510 GOSUB 4190
4520 FOR T=1 TO 500
4530 NEXT T
4540 CALL CLEAR
4550 END

```

Nelle linee 210-550 sono definiti tutti i caratteri speciali per ottenere gli assi cartesiani ed i punti che formeranno il grafico. In 560 è definita la funzione che è l'espressione analitica della parabola. Il titolo occupa la parte seguente fino alla linea 980. Chi volesse ridurre tutto all'essenziale, senza nulla concedere all'estetica, può sostituire questa parte con la semplice istruzione `PRINT TAB(10);"PARABOLE" : GOTO 1100`.

Il menù (990-1100) prevede il salto a tre sezioni di lavoro ed uno alla fine del programma; la scelta è accettata con l'istruzione `CALL KEY`.

La prima sezione occupa le linee da 1110 a 1660. L'input dei dati avviene in una routine posta alla riga 2700. Il carattere 35 che appare nell'istruzione 1160 è stato ridefinito in modo da rappresentare l'esponente della variabile X nel termine di secondo grado. DELTA rappresenta il discriminante $b^2 - 4ac$, VX e VY sono le coordinate del vertice, VF è l'ordinata del fuoco (l'ascissa è la stessa del vertice) e, proseguendo, YD è l'ordinata dei punti che costituiscono la direttrice. Al termine di ogni calcolo c'è la stampa dei risultati. A questo punto lo schermo è quasi completamente occupato e si rimanda alla routine di riga 4010 per inviare il messaggio `PREMI 1 PER CONTINUARE`. Si prosegue con le intersezioni della parabola con gli assi; C fornisce l'intersezione con l'asse Y (1350), mentre, perché esistano intersezioni con l'asse X, è necessario che $\Delta = 0$ (linea 1360); X1 e X2 contengono le soluzioni della equazione di secondo grado $ax^2 + bx + c = 0$ e pertanto forniscono i punti di intersezione con l'asse orizzontale. Per ottenere il grafico della parabola è necessario rispondere affermativamente nell'input di linea 1430. La routine 4290 traccia gli assi cartesiani; la variabile SCALE è posta inizialmente uguale ad 1, successivamente la routine 2430 disegna il grafico. I salti condizionati alle righe 1490 e 1500 hanno effetto solo se si giunge in quest'area di lavoro provenendo dalle sezioni 2 e 3, altrimenti si prosegue con le due istruzioni `GOSUB`. La prima provoca un breve commento musicale che può essere omissivo, la seconda fa apparire il messaggio per proseguire. Le linee seguenti servono a cambiare il fattore di scala. La variabile SCALE rappresenta il valore di ogni intervallo segnato sugli assi cartesiani; inizialmente tale valore è posto uguale ad 1. Se $SCALE=2$, ogni trattino rappresenta due unità, pertanto, se in un primo tempo il vertice della parabola cade fuori campo, ponendo il fat-

tore di scala uguale a 2, a 3, ecc., l'immagine verrà rimpicciolita ed il vertice della parabola potrà essere rappresentato all'interno dello schermo. Viceversa, quando si studia l'intersezione con una retta, se l'eccessiva vicinanza dei punti creasse confusione, ponendo il fattore di scala uguale a 0.5 o 0.2, si ottiene un'immagine ingrandita e più chiara in certi dettagli. Unica limitazione, deve essere SCALA > 0. Si consiglia di cambiare i valori con gradualità fino ad ottenere il grafico più soddisfacente.

È interessante notare che il grafico ridotto o ingrandito della parabola si ottiene semplicemente ponendo $A = A \cdot \text{SCALA}$ e $C = C / \text{SCALA}$ (linee 1620-1630).

La seconda parte tratta l'intersezione tra una parabola ed una retta. I coefficienti sono, a scelta, accettati in forma frazionaria o decimale ed è previsto che la retta sia parallela all'asse Y, nel qual caso la sua equazione è $x=k$. La variabile VERT assume il valore 1 quando la retta è verticale. Per trovare i punti di intersezione si risolve il sistema formato dalle equazioni della parabola e della retta. Sottraendo membro a membro si ottiene la seguente equazione di secondo grado:

$$ax^2 + (b-m)x + c-q = 0$$

dove m e q sono i coefficienti dell'equazione della retta.

Il discriminante è calcolato alla linea 2030 e, se questo non è negativo, si determinano le coordinate dei punti di intersezione (istruzioni 2080-2110). Per il grafico si rimanda alla linea 1430. Prima viene disegnata la parabola e, dopo aver posto $A=0$, $B=M$ e $C=Q$, utilizzando lo stesso ciclo di stampa, si disegna la retta; infatti, togliendo dall'equazione della parabola il termine di secondo grado, si ottiene l'equazione di una retta. Alla linea 2960 inizia lo studio grafico ed analitico di una parabola passante per tre punti dati. Anche in questo caso si possono inserire le coordinate in forma frazionaria. Nella presentazione del programma si è già parlato di come vengono calcolate le costanti a , b , c che compaiono nella equazione della parabola. Per risolvere il sistema viene usata la regola di Cramer. Alla linea 3990 si calcola il valore di un determinante di terzo ordine; DD è il determinante del sistema, mentre DD1, DD2 e DD3 sono i determinanti che si ottengono da questo sostituendo i termini noti agli elementi della colonna formata con i coefficienti dell'incognita che si considera.

A, B e C sono calcolate alla linea 3480 con una approssimazione a meno di un millesimo e, se si desidera proseguire con lo studio della parabola, il programma torna alla prima sezione.

Le linee 3610 e seguenti sono occupate da diverse routine. La prima di queste è piuttosto una sottosezione dedicata all'input dei dati ed alla stampa di una retta parallela all'asse delle Y. In base al valore di DF

(3790) la retta occupa quattro diverse sottocolonne all'interno di ogni colonna, in questo modo gli errori dovuti all'approssimazione sono assorbiti dal graficismo.

Delle altre subroutine si è già parlato nella descrizione del programma e non è necessario commentarle ulteriormente.

Ancora matematica

Studio di una funzione e sue derivate

Lo studio di una funzione consiste nel determinarne l'insieme di esistenza, nel trovare eventuali intersezioni con gli assi, nella ricerca degli asintoti, nel determinare gli intervalli dove $f(x)$ è crescente o decrescente ed i massimi e minimi relativi, infine calcolarne il valore in corrispondenza di alcuni valori della variabile x .

Il programma si limita allo studio di quelle che sono le più semplici tra le funzioni razionali, cioè i polinomi.

La loro forma è pertanto:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

Il programma è diviso in quattro sezioni: (1) calcolo di $y=f(x)$ per determinati valori di x ; (2) calcolo delle derivate; (3) determinazione della retta tangente alla curva in un punto; (4) ricerca degli intervalli dove la funzione è crescente e decrescente e dei punti di massimo e di minimo relativo.

La ricerca dei punti di intersezione con l'asse x può essere svolta con il prossimo programma.

Inizialmente bisogna indicare il grado di un polinomio ed i coefficienti delle potenze decrescenti della x ; se manca una potenza della x il corrispondente coefficiente è 0.

Alcune tecniche di programmazione, per calcolare il valore del polino-

mio, fanno ricorso all'algoritmo di Horner-Ruffini; tuttavia, in questa sede, per una più immediata comprensione, si è preferito procedere nel modo consueto e sommare i prodotti di ogni coefficiente per la corrispondente potenza di x .

Per il calcolo della derivata in un punto si ricorre allo stesso metodo dopo aver trovato i coefficienti della funzione derivata. Come è noto

$$D ax^n = anx^{n-1}$$

In modo analogo si ricavano i coefficienti delle derivate di ordine superiore.

Ad esempio:

$$f(x) = 3x^2 - 5x + 7$$

$$f'(x) = 6x - 5$$

$$f''(x) = 6$$

La derivata della funzione in un punto x_0 rappresenta il coefficiente angolare della tangente alla curva di equazione $y=f(x)$ nel punto di ascissa x_0 . Per trovare l'equazione della tangente in un punto x_0 , si calcola $f(x_0)$ e $f'(x_0)$; l'equazione richiesta risulta

$$y - f(x_0) = f'(x_0) \times (x - x_0)$$

ed in forma esplicita:

$$y = f'(x_0)x + f'(x_0)x_0 + f(x_0)$$

dove $m = f'(x_0)$ e $q = f'(x_0)x_0 + f(x_0)$.

L'ultima sezione, nella quale vengono calcolati massimi e minimi è alquanto più complessa.

La ricerca viene effettuata all'interno di un intervallo definito dall'utente, a partire dall'estremo inferiore e per valori discreti ottenuti incrementando successivamente la x di una quantità dx , anch'essa definita dall'utente.

Il programma procede finché non si verifica una delle seguenti eventualità: (1) $f'(x)=0$, cioè la derivata prima si annulla per un dato valore di x ; (2) le derivate, calcolate in due punti successivi, sono discordi.

Nel primo caso, detto x_0 il punto in cui si annulla la derivata, si calcolano le derivate successive fino a trovare quella che in x_0 non si annulla. Se quest'ultima è di ordine pari, in corrispondenza di x_0 c'è un massimo relativo se la derivata è negativa e un minimo relativo se è positiva; invece, se è di ordine dispari, in x_0 la funzione non ha né massimo né minimo.

Nel secondo caso in cui cambi il segno della derivata calcolata in due punti successivi, è evidente che tra questi deve esistere un punto x_0 in cui $f'(x_0)=0$. Per trovare x_0 si utilizza la stessa subroutine per la ricerca nell'intervallo principale, dopo aver posto gli estremi uguali a $x-dx$ e x e l'incremento uguale a un decimo del precedente. Nuovamente, o si trova il punto x_0 tale che $f'(x_0)=0$ o si trova un intervallo più piccolo (1/10 del precedente) in cui si trova x_0 . Il procedimento viene ripetuto cinque volte fino a trovare la soluzione con una sufficiente approssimazione. Se l'incremento iniziale è 1, x_0 è calcolato con l'approssimazione a meno di un centomillesimo. È possibile aumentare la precisione aumentando il numero di iterazioni.

```

100 REM  FUNZIONI RAZIONALI
110 REM  *****
120 REM  di: Sergio Borsani
130 REM  versione: TI BASIC
140 CALL CLEAR
150 DIM A(20),B(20),F(20)
160 CALL CHAR(130,"7848484878")
170 CALL CHAR(131,"2060202020")
180 CALL CHAR(132,"7808780878")
190 CALL CHAR(133,"7808780878")
200 CALL CHAR(134,"4040507810")
210 CALL CHAR(135,"7840780878")
220 CALL CHAR(136,"7840784878")
230 CALL CHAR(137,"7808080808")
240 CALL CHAR(138,"7848784878")
250 CALL CHAR(139,"7848780878")
260 CALL CHAR(135,"FFFFFFFFFFFFFF")
270 CALL COLOR(16,7,1)
280 REM
290 REM      TITOLO
300 REM
310 PRINT TAB(11);"FUNZIONI":
320 PRINT TAB(7);"RAZIONALI INTERE":
330 CALL HCHAR(10,6,155,22)
340 CALL VCHAR(11,6,155,7)
350 CALL VCHAR(11,27,155,7)
360 CALL HCHAR(18,6,155,22)
370 FOR T=1 TO 1000
380 NEXT T
390 GOSUB 1710
400 CALL CLEAR
410 PRINT "PREMI:":
420 PRINT "1. PER IL VALORE DELLA FUN-":
430 PRINT "2. PER IL VALORE DELLE DERI-":
440 PRINT "3. PER L'EQUAZ. DELLA RETTA":
450 PRINT "4. PER LO STUDIO DELLA FUN-":
460 PRINT "5. PER FINIRE":
470 CALL KEY(0,K,S)
480 IF S=0 THEN 470
490 IF K<49 THEN 470
500 IF K>53 THEN 470
510 ON K-48 GOTO 520,750,980,1250,1620
520 CALL CLEAR
530 PRINT "CALCOLO DELLA FUNZIONE:":
540 PRINT "F(X)=":
550 FOR I=N TO 1 STEP -1
560 IF A(I)=0 THEN 600
570 PRINT STR$(A(I));"X";
580 IF I=1 THEN 600
590 GOSUB 2690
600 IF A(I-1)<=0 THEN 630
610 IF (I=1)*(A(0)=0) THEN 640
620 PRINT "+";
630 NEXT I
640 IF A(0)=0 THEN 660
650 PRINT STR$(A(0)):
660 PRINT
670 INPUT "X = ":X
680 PRINT
690 GOSUB 1920
700 PRINT "F(";STR$(X);")=";F(0)
710 PRINT ":
720 INPUT "VUOI IL CALCOLO PER UN ALTRO PUNTO? (S/N) ":R$

```

```

730 IF SEG$(R$,1,1)="S" THEN 660
740 GOTO 400
750 CALL CLEAR
760 PRINT "VALORE DELLA DERIVATA NEL"
770 PRINT "PUNTO X"
780 INPUT "X = ":X
790 GOSUB 2770
800 K=0
810 K=K+1
820 IF K>N THEN 920
830 F(K)=0
840 GOSUB 2000
850 PRINT "DERIVATA"
860 PRINT "F";STR$(K);"(";STR$(X);")=";F(K)
870 PRINT " "
880 PRINT " "
890 IF K=N THEN 930
900 INPUT "VUOI UNA DERIVATA DI ORDINE SUPERIORE? (S/N) ":R$
910 IF SEG$(R$,1,1)="S" THEN 810
920 PRINT " "
930 INPUT "VUOI DERIVARE IN UN ALTRO PUNTO? (S/N) ":R$
940 IF SEG$(R$,1,1)="S" THEN 750 ELSE 400
950 REM
960 REM TANGENTE
970 REM
980 CALL CLEAR
990 PRINT "EQUAZIONE DELLA RETTA"
1000 PRINT "TANGENTE IN CORRISPONDENZA"
1010 PRINT "DI UN PUNTO"
1020 PRINT " "
1030 INPUT "X = ":X
1040 GOSUB 2770
1050 K=1
1060 F(K)=0
1070 GOSUB 1920
1080 GOSUB 2000
1090 PRINT " "
1100 M=INT(100*F(1))/100
1110 Y1=INT(100*F(0))/100
1120 Q=-M*X+Y1
1130 PRINT "Y="
1140 IF M=0 THEN 1160
1150 PRINT STR$(M);"X"
1160 IF (M=0)*(Q=0) THEN 1190
1170 IF Q=0 THEN 1220
1180 IF Q>0 THEN 1210
1190 PRINT Q
1200 GOTO 1220
1210 PRINT "+";STR$(Q)
1220 PRINT " "
1230 INPUT "VUOI IL CALCOLO IN UN ALTRO PUNTO? (S/N) ":R$
1240 IF SEG$(R$,1,1)="S" THEN 1020 ELSE 400
1250 CALL CLEAR
1260 PRINT "IN QUESTA SEZIONE SI STUDIA"
1270 PRINT "LA FUNZIONE IN UN INTERVALLO"
1280 PRINT "SCRIVERE GLI ESTREMI E"
1290 PRINT "L'INCREMENTO DELLA X"
1300 INPUT "X1,X2 ":INIZ,FINE
1310 PRINT " "
1320 INPUT "INCREMENTO = ":INCR
1330 X=INIZ
1340 GOSUB 1920
1350 FX1=F(X)
1360 X=FINE
1370 GOSUB 1920
1380 FX2=F(X)
1390 CALL CLEAR
1400 PRINT "PUNTO", "FUNZIONE"
1410 PRINT "-----"
1420 PRINT INIZ,FX1
1430 X=INIZ
1440 X2=FINE
1450 DX=INCR
1460 GOSUB 2090
1470 IF SD2=0 THEN 1520
1480 IF SD1>0 THEN 1510
1490 PRINT "DECRESCENTE"
1500 GOTO 1520
1510 PRINT "CRESCENTE"
1520 PRINT FINE,FX2
1530 PRINT "-----"
1540 PRINT " "
1550 PRINT "PREMI UN TASTO"
1560 CALL KEY(0,K,S)
1570 IF S=0 THEN 1560
1580 GOTO 400

```

```

1590 REM
1600 REM     FINE
1610 REM
1620 CALL CLEAR
1630 PRINT TAB(13);"FINE":;
1640 FOR T=1 TO 1000
1650 NEXT T
1660 CALL CLEAR
1670 END
1680 REM
1690 REM     INPUT DEI DATI
1700 REM
1710 CALL CLEAR
1720 PRINT "LA FUNZIONE DEVE ESSERE UN POLINOMIO ORDINATO SECONDO"
1730 PRINT "LE POTENZE DECRESCENTI DELLAVARIABILE INDIPENDENTE":
1740 INPUT "GRADO DEL POLINOMIO ";N
1750 IF N<=20 THEN 1790
1760 PRINT
1770 PRINT "VALORE TROPPO ALTO, RIPROVA":
1780 GOTO 1740
1790 PRINT
1800 PRINT "COEFFICIENTI:":
1810 FOR I=N TO 0 STEP -1
1820 PRINT "  DI X^";
1830 GOSUB 2690
1840 PRINT "  =";
1850 INPUT A(I)
1860 NEXT I
1870 RETURN
1880 REM
1890 REM     VALORE DELLA
1900 REM     FUNZIONE
1910 REM
1920 F(0)=0
1930 FOR I=0 TO N
1940 F(0)=F(0)+A(I)*X^I
1950 NEXT I
1960 RETURN
1970 REM
1980 REM     DERIVATE
1990 REM
2000 FOR J=0 TO N-K
2010 B(J)=B(J+1)*(J+1)
2020 F(K)=F(K)+B(J)*X^J
2030 NEXT J
2040 RETURN
2050 REM
2060 REM     RICERCA
2070 REM     MASSIMI E MINIMI
2080 REM
2090 V=0
2100 K=1
2110 F(1)=0
2120 X=X1
2130 GOSUB 2770
2140 GOSUB 2000
2150 SD1=SGN(F(1))
2160 IF SD1=0 THEN 2300
2170 FOR I=X1+DX TO X2 STEP DX
2180 F(1)=0
2190 GOSUB 2770
2200 GOSUB 2000
2210 SD2=SGN(F(1))
2220 IF SD2=0 THEN 2300
2230 IF SD1<>SD2 THEN 2520
2240 SD1=SD2
2250 NEXT X
2260 RETURN
2270 REM
2280 REM     DERIV. I=0
2290 REM
2300 GOSUB 2770
2310 FOR K=1 TO N
2320 F(K)=0
2330 GOSUB 2000
2340 IF K=1 THEN 2440
2350 IF F(K)=0 THEN 2440
2360 IF K/2<INT(K/2) THEN 2450
2370 IF F(K)>0 THEN 2410
2380 PRINT "CRESCENTE":
2390 PRINT "  MASSIMO REL.":
2400 GOTO 2450
2410 PRINT "DECRESCENTE":
2420 PRINT "  MINIMO REL.":
2430 GOTO 2450
2440 NEXT K

```

```

2450 X1=X+INCR
2460 K2=FINE
2470 DX=INCR
2480 GOTO 2100
2490 REM
2500 REM SD1<>SD2
2510 REM
2520 V=V+1
2530 X1=X-DX
2540 X2=X
2550 DX=DX/10
2560 IF V>5 THEN 2580
2570 GOTO 2100
2580 IF SD1=0 THEN 2620
2590 PRINT "DECRESCENTE":
2600 PRINT "MINIMO REL.":
2610 GOTO 2640
2620 PRINT "CRESCENTE":
2630 PRINT "MASSIMO REL.":
2640 V=0
2650 GOTO 2450
2660 REM
2670 REM ESPONENTI
2680 REM
2690 FOR H=1 TO LEN(STR$(I))
2700 ESP=VAL(SEG$(STR$(I),H,1))
2710 PRINT CHR$(130+ESP);
2720 NEXT H
2730 RETURN
2740 REM
2750 REM B(I)=A(I)
2760 REM
2770 FOR I=0 TO N
2780 B(I)=A(I)
2790 NEXT I
2800 RETURN

```

Nelle righe iniziali sono definiti i caratteri per rappresentare gli esponenti; per ragioni mnemoniche l'ultima cifra del numero del carattere è uguale all'esponente specifico, pertanto 130 è il numero del carattere che rappresenta l'esponente 0, 131 corrisponde all'esponente 1, e così via.

I coefficienti del polinomio sono accettati nella subroutine 1710 e memorizzati nel vettore A(I); N è invece il grado del polinomio.

Le istruzioni 540-650 fanno apparire la funzione sullo schermo. Gli esponenti vengono scritti dalla routine 2690-2730. Con qualche condizione in più si evita di scrivere il coefficiente se questo è uguale a zero o di scrivere l'esponente uno. Il calcolo del valore della funzione in un punto X è svolto dalla routine 1920. Per il calcolo delle derivate si trasferiscono i coefficienti in B(I) per non perderli e si passa alla GOSUB 2000. È inteso che il valore della funzione è assunto da F(0), quello della derivata prima da F(1), quello della derivata seconda da F(2), e così via.

Per scrivere l'equazione esplicita della retta tangente al grafico della funzione in un punto X si calcolano il valore della funzione e della sua derivata prima in quel punto e successivamente il coefficiente angolare M (linea 1100) e l'ordinata all'origine Q (linea 1120). Alle linee 1130-1210 viene scritta l'equazione esplicita della retta; anche qui qualche accorgimento è volto a dare una scrittura corretta, infatti non verrà scritto il termine corrispondente se M o Q dovessero risultare uguali a zero, ma apparirà la scrittura $Y=0$ se entrambi lo fossero.

La parte più impegnativa è senza dubbio quella concernente la ricerca dei massimi e dei minimi relativi. L'algoritmo seguito è esattamente quello illustrato nella parte introduttiva qualora venga trovato un punto

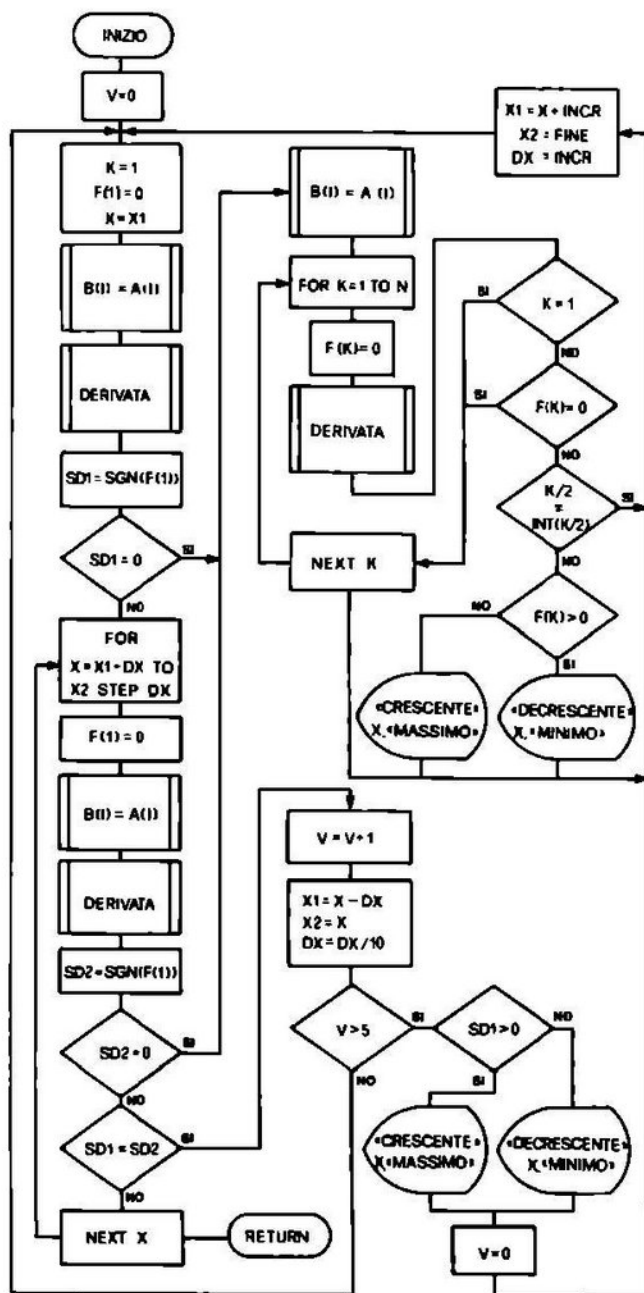


Fig. 10

dove la derivata prima è uguale a zero. SD1 ed SD2 sono le variabili che contengono il segno della derivata calcolata rispettivamente nei punti X e $X + DX$. Se per un dato incremento della X non si verifica la condizione $SD1 = 0$ oppure $SD2 = 0$, allora si guarda se SD1 sia diverso da SD2 poiché l'inversione di segno della derivata indica che siamo in presenza almeno di un massimo o di un minimo relativo. Si procede nel seguente modo: se nell'intervallo principale ove si studia la funzione viene trovato un intervallino ai cui estremi le derivate assumono segni opposti, si esplora l'intervallino con un incremento della variabile indipendente dieci volte minore del precedente fino a trovare o un punto ove la derivata prima si annulla oppure un altro intervallino ai cui estremi le derivate hanno segno opposto. Il ciclo viene ripetuto fino ad un massimo di cinque volte (variabile V), ad ogni iterazione il risultato viene trovato con un'approssimazione che aumenta di un ordine di grandezza e nulla vieta di proseguire in questo modo fino ad ottenere l'approssimazione desiderata.

Data la complessità di questa parte viene riportato il diagramma a blocchi relativo (figura 10).

Vediamo ora un esempio di applicazione pratica. È stato scelto un polinomio di terzo grado ($N=3$) e in input sono stati scritti i seguenti coefficienti: 4, -72, 324, 0. Premendo il tasto 1 al menù appare la funzione scritta nel modo consueto; per $x=3$ viene calcolato il valore $f(x)=432$. Passando, con il tasto 4 del menù, allo studio della funzione, viene chiesto di specificare gli estremi dell'intervallo e l'incremento della X con il quale esso deve essere esplorato. Sono stati scelti gli estremi 0 e 10, l'incremento è stato posto uguale a 0.5. Come si vede dalla figura (esempio 6) vengono dati i valori della funzione agli estremi dell'intervallo, un punto di massimo relativo per $x=3$ e un punto di minimo relativo per $x=9$, inoltre vengono specificati gli intervalli in cui la funzione è crescente o decrescente.

Quando sussistono dei dubbi è preferibile interrogare nuovamente il computer esplorando l'intervallo con un altro valore dell'incremento, tracciando il grafico per punti e calcolando il valore della derivata in corrispondenza di alcuni di essi.

RUN

CALCOLO DELLA FUNZIONE:

$$F(X) = 4X^3 - 72X^2 + 324X$$

$$X = 3$$

$$F(3) = 432$$

VUOI IL CALCOLO IN UN ALTRO
PUNTO? (S/N) ■

IN QUESTA SEZIONE SI STUDIA
LA FUNZIONE IN UN INTERVALLO

SCRIVERE GLI ESTREMI E
L'INCREMENTO DELLA X

$$X1, X2 \quad 0, 10$$

$$\text{INCREMENTO} = .5$$

PUNTO	FUNZIONE
0	0
3	CRESCENTE MASSIMO REL.
9	DECRESCENTE MINIMO REL.
10	CRESCENTE 40

* PREMI UN TASTO

Soluzione delle equazioni algebriche intere

Le equazioni algebriche intere sono quelle equazioni che possono essere espresse nella forma:

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0 = 0$$

Convenzionalmente il grado massimo è stato posto uguale a 20.

Il programma fornisce solo, se esistono, le soluzioni reali dell'equazione. I metodi adottati differiscono al variare del grado dell'equazione. Se il grado è 1, l'equazione è del tipo $ax + b = 0$ e la soluzione è $x = -b/a$. Se l'equazione è di secondo grado è del tipo $ax^2 + bx + c = 0$ ed il suo determinante è $D = b^2 - 4ac$. Se risulta $D < 0$ l'equazione non ammette soluzioni reali; se $D = 0$, ammette due soluzioni coincidenti; infine, se

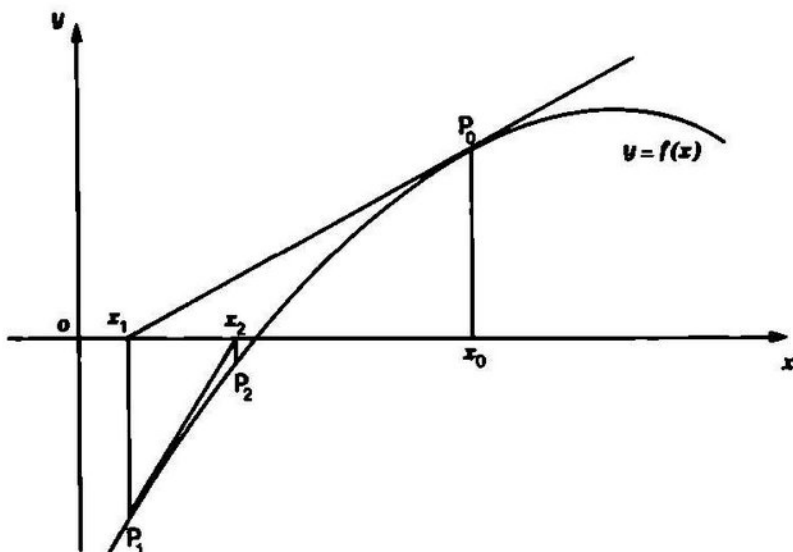


Fig. 11

$D > 0$ esistono due soluzioni distinte date dalle formule:

$$x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad ; \quad x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Se il grado dell'equazione è maggiore o uguale a 3 si segue il metodo di Newton. Questo consiste in una serie di successive approssimazioni. Nella figura 11 viene mostrato il grafico di una generica funzione $f(x)$ ed una retta tangente alla curva nel punto P_0 di ascissa x_0 . Come è noto la derivata della funzione nel punto P_0 rappresenta il coefficiente angolare della tangente. Detto x_1 il punto di intersezione della tangente con l'asse delle x , è $f'(x_0) = f(x_0)/(x_0 - x_1)$ e da qui si ricava il valore di x_1 :

$$x_1 = x_0 - f(x_0)/f'(x_0)$$

Il metodo consiste nell'assumere x_1 come approssimazione della radice reale r e da questo calcolare il valore x_2 in modo del tutto analogo:

$$x_2 = x_1 - f(x_1)/f'(x_1)$$

Poi il valore di x_3 , e così via, fino a trovare la soluzione con l'approssimazione desiderata. Generalmente bastano poche iterazioni per trovare una soluzione.

Nell'eventualità che sia $f'(x_0)=0$ la tangente non ha intersezioni con l'asse delle x e il programma è fornito di una funzione random per fissare un nuovo punto di partenza per le iterazioni.

In ogni caso, trovata una radice reale r dell'equazione, detto $P(x)$ il polinomio al primo membro dell'eguaglianza, deve essere $P(x) = Q(x) \times (x-r)$ e si può procedere al calcolo dei coefficienti del nuovo polinomio $Q(x)$ con la regola di Ruffini. Ad esempio, data l'equazione $x^3 - 4x^2 + x + 6 = 0$ e trovata con il metodo di Newton la soluzione $x=2$, applicando la regola di Ruffini si trovano i coefficienti del polinomio quoziente $Q(x)$.

$$\begin{array}{r|rrrr} 2 & 1 & -4 & 1 & 6 \\ & & 2 & -4 & -6 \\ \hline & 1 & -2 & -3 & 0 \end{array}$$

Le soluzioni dell'equazione di secondo grado $x^2 - 2x - 3 = 0$ sono anche soluzione dell'equazione di terzo grado data inizialmente. Ogni volta che viene individuata una radice dell'equazione è possibile scrivere un'equazione di grado immediatamente inferiore. Il programma utilizza il metodo di Newton per trovare soluzioni e per abbassare conseguentemente il grado dell'equazione finché non si arriva al secondo, dopo di che si usano le formule per la soluzione di equazioni di secondo grado.

Un breve commento merita ancora il modo in cui viene trovato il valore del polinomio nella routine 2150.

L'algoritmo utilizzato è quello di Horner-Ruffini che, per quanto sia meno familiare, consente al computer un calcolo più veloce con un minor numero di istruzioni.

Il polinomio:

$$P(x) = a_n x^n + \dots + a_2 x^2 + a_1 x + a_0$$

può essere scritto nella forma:

$$((((a_n x + a_{n-1})x + \dots)x + a_2)x + a_1)x + a_0$$

Per calcolare il valore del polinomio per $x=x_0$, si definiscono i seguenti numeri:

$$\begin{aligned} k_0 &= a_n \\ k_1 &= k_0 x_0 + a_{n-1} \\ &\dots\dots\dots \\ k_{n-1} &= k_{n-2} x_0 + a_1 \\ k_n &= k_{n-1} x_0 + a_0 \end{aligned}$$

risulta:

$$P(x_0) = k_0$$

In modo del tutto analogo viene calcolato il valore della derivata nel punto x_0 .

Per gli esponenti, i segni di radice e di frazione sono stati definiti simboli speciali in modo da ottenere una notazione uguale a quella convenzionalmente adottata in matematica. Come è noto, il computer, senza istruzioni particolari, scrive, le potenze nella forma X^N , le radici quadrate nel modo $SQR(N)$ e, per il segno di frazione, usa il simbolo $/$.

Quando il programma si prefigge anche uno scopo didattico, non credo sia ingiustificato impiegare una certa quantità di memoria per ottenere una maggior chiarezza espositiva.

```

100 REM      SOLUZIONI DELLE
110 REM      EQUAZIONI ALGEBRICHE
120 REM      *****
130 REM      di Sergio Borsani
140 REM      versione: TI BASIC
150 CALL CLEAR
160 PRINT "      EQUAZIONI ALGEBRICHE":::
170 DIM A(20),B(20),R(20)
180 REM
190 REM      ESPONENTI
200 REM
210 CALL CHAR(128,"7B4B4B4B7B")
220 CALL CHAR(129,"2060202020")
230 CALL CHAR(130,"7B0B7B0B7B")
240 CALL CHAR(131,"7B0B7B0B7B")
250 CALL CHAR(132,"6060507B10")
260 CALL CHAR(133,"7B607B0B7B")
270 CALL CHAR(134,"7B607B6B7B")
280 CALL CHAR(135,"7B9B0B0B0B")
290 CALL CHAR(136,"7B6B7B6B7B")
300 CALL CHAR(137,"7B6B7B0B7B")
310 CALL CHAR(141,"506B4B4B4B")
320 REM
330 REM      SIMBOLO DI RADICE
340 REM
350 CALL CHAR(138,"0101026214140B0B")
360 CALL CHAR(139,"00000000000000FF")
370 REM
380 REM      SEGNO DI FRAZIONE
390 REM
400 CALL CHAR(140,"00000000FF")
410 CALL CLEAR
420 AP=0
430 PRINT "      IL PROGRAMMA PERMETTE LA"
440 PRINT "SOLUZIONE DELLE EQUAZIONI"
450 PRINT "AVANTI LA FORMA:::"
460 PRINT "A X";CHR$(141);"+ ... +A X";CHR$(130);"+A X+A <0"
470 PRINT TAB(3);CHR$(141);TAB(14);CHR$(130);TAB(19);CHR$(129);
475 PRINT TAB(23);CHR$(128):::
480 INPUT "GRADO DELL'EQUAZIONE = ":G
490 PRINT :::
500 IF G=0 THEN 540
510 PRINT
520 PRINT "ERRORE!":
530 GOTO 480
540 IF G<=20 THEN 580
550 PRINT "IL GRADO E' TROPPO ALTO"
560 PRINT "BISOGNA MODIFICARE L'ISTRU-": "ZIONE DIM":
570 GOTO 480
580 PRINT "SCRIVI I COEFFICIENTI DELLE"
590 PRINT "POTENZE DECRESCENTI DELLA X":
600 PRINT "IN MANCANZA DI UN TERMINE"
610 PRINT "PONI A=0":
620 PRINT "COEFFICIENTI:":
630 N=G
640 FOR I=0 TO N
650 PRINT "A(ISTR$(N-I)) = "

```

```

660 INPUT A(N-I)
670 IF A(N)<>0 THEN 700
680 PRINT "ERRORE!":
690 GOTO 690
700 NEXT I
710 CALL CLEAR
720 PRINT "L'EQUAZIONE DI ";STR$(N);CHR$(128); " GRADO E':::
730 FOR J=N TO 0 STEP -1
740 IF A(I)=0 THEN 860
750 IF A(I)<0 THEN 780
760 IF I=N THEN 780
770 PRINT "+";
780 PRINT STR$(A(I));
790 IF I=0 THEN 860
800 PRINT "X";
810 IF I=1 THEN 860
820 FOR K=1 TO LEN(STR$(I))
830 L=ASC(STR$(I),K,1)
840 PRINT CHR$(80+L);
850 NEXT K
860 NEXT I
870 PRINT "=0":
880 IF N=1 THEN 910
890 IF N=2 THEN 950
900 IF N=3 THEN 1480
910 PRINT "SOLUZIONE:":
920 PRINT "X =";-A(0)/A(1)
930 PRINT :
940 GOTO 2640
950 IF A(0)=0 THEN 1440
960 GOSUB 2130
970 GOSUB 2460
980 IF D=0 THEN 1030
990 IF D>0 THEN 1160
1000 PRINT "L'EQUAZIONE NON AMMETTE"
1010 PRINT "RADICI REALI.":
1020 GOTO 2640
1030 PRINT "L'EQUAZIONE AMMETTE "
1040 PRINT "DUE RADICI COINCIDENTI":
1050 PRINT TAB(7);-B(1)
1060 IF -B(1)>2*B(2) THEN 1090
1070 LU=LEN(STR$(2*B(2)))
1080 GOTO 1100
1090 LU=LEN(STR$(-B(1)))
1100 PRINT "X1=X2=";
1110 PRINT TAB(9+LU);"=";XX1
1120 CALL HCHAR(23,10,140,LU)
1130 PRINT TAB(7);2*B(2)
1140 PRINT :
1150 GOTO 2640
1160 PRINT "SOLUZIONI:":
1170 LD=LEN(STR$(D))
1180 LB=LEN(STR$(-B(1)))
1190 LA=LEN(STR$(2*B(2)))
1200 IF 3+LB+LD>LA THEN 1230
1210 LU=LA
1220 GOTO 1240
1230 LU=3+LB+LD
1240 FOR J=1 TO 2
1250 PRINT TAB(5);-B(1);
1260 IF J=2 THEN 1290
1270 PRINT "-";
1280 GOTO 1300
1290 PRINT "+";
1300 CALL HCHAR(23,11+LB,139,LD+1)
1310 PRINT TAB(9+LB);D
1320 CALL HCHAR(23,10+LB,138)
1330 PRINT "X";STR$(J);"=";
1340 PRINT TAB(8+LU);"=";
1350 IF J=2 THEN 1380
1360 PRINT XX1
1370 GOTO 1390
1380 PRINT XX2
1390 CALL HCHAR(23,7,140,LU+2)
1400 PRINT TAB(5);2*B(2):
1410 NEXT J
1420 PRINT
1430 GOTO 2640
1440 PRINT "SOLUZIONI:":
1450 PRINT "X1 = 0":
1460 PRINT "X2 =";-A(1)/A(2):
1470 GOTO 2640
1480 CTR=0
1490 GOSUB 2130
1500 CTR=CTR+1
1510 IF B(0)<>0 THEN 1630

```

```

1520 R(CTR)=0
1530 XX=0
1540 GOSUB 2560
1550 IF N<>2 THEN 1500
1560 GOSUB 2460
1570 IF D<0 THEN 2010
1580 CTR=CTR+1
1590 R(CTR)=XX1
1600 CTR=CTR+1
1610 R(CTR)=XX2
1620 GOTO 2010
1630 X0=0
1640 GOSUB 2310
1650 IF P=0 THEN 1720
1660 IF P1=0 THEN 1760
1670 IF T>50 THEN 1930
1680 PRINT ":::
1690 PRINT "SPIACENTE, NON RIESCO A": "TROVARE LE SOLUZIONI"
1700 PRINT "QUASI CERTAMENTE L'EQUAZIONE": "NON AMMETTE SOLUZIONI REALI.": ":::
1710 GOTO 2640
1720 R(CTR)=X0
1730 XX=X0
1740 GOSUB 2560
1750 IF N=2 THEN 1560 ELSE 1500
1760 RANDOMIZE
1770 INCR=INT(RND*10)+1
1780 IF INCR/2=INT(INCR/2) THEN 1810
1790 X0=X0-INCR
1800 GOTO 1820
1810 X0=X0+INCR
1820 VOLTE=VOLTE+1
1830 IF VOLTE=3 THEN 1850
1840 GOTO 1640
1850 PRINT "SONO IN DIFFICOLTA' PER FA-"
1860 PRINT "VORE INDICA UN VALORE DI X.": :::
1870 PRINT "SE NON VUOI PROSEGUIRE"
1880 PRINT "SCRIVI 9999": :::
1890 INPUT "X = ":X0
1900 PRINT ":::
1910 VOLTE=0
1920 IF X0=9999 THEN 1690 ELSE 1510
1930 X0=X
1940 GOSUB 2210
1950 IF ABS(P)>1E-10 THEN 1760
1960 AP=1
1970 R(CTR)=X0
1980 XX=X0
1990 GOSUB 2560
2000 GOTO 1550
2010 IF AP=1 THEN 2040
2020 PRINT "SOLUZIONI:": :::
2030 GOTO 2050
2040 PRINT "SOLUZIONI APPROSSIMATE": :::
2050 FOR I=1 TO CTR
2060 PRINT "X";STR$(I); " =";R(I)
2070 NEXT I
2080 PRINT ":::
2090 GOTO 2640
2100 REM
2110 REM B(I)=A(I)
2120 REM
2130 FOR I=0 TO N
2140 B(I)=A(I)
2150 NEXT I
2160 RETURN
2170 REM
2180 REM VALORE POLINOMIO
2190 REM E DERIVATA IN X0
2200 REM
2210 P=B(N)
2220 P1=0
2230 FOR I=N-1 TO 0 STEP -1
2240 P1=P1*X0+P
2250 P=P*X0+B(I)
2260 NEXT I
2270 RETURN
2280 REM
2290 REM METODO DI NEWTON
2300 REM
2310 T=0
2320 T=T+1
2330 IF T>50 THEN 2410
2340 GOSUB 2210
2350 IF P=0 THEN 2410
2360 IF P1=0 THEN 2410
2370 X=X0-P/P1

```



```

2380 REM PUOI STAMPARE X0,P
2390 X0=X
2400 GOTO 2720
2410 RETURN
2420 REM
2430 REM SOLUZIONE EQ.
2440 REM DI SECONDO GRADO
2450 REM
2460 D=B(1) 2-4*B(2)*B(0)
2470 IF D<0 THEN 2490
2480 GOTO 2510
2490 XX1=(-B(1)-SOR(D))/(2*B(2))
2500 XX2=(-B(1)+SOR(D))/(2*B(2))
2510 RETURN
2520 REM
2530 REM COEFF. POLINOMIO
2540 REM QUOZIENTE
2550 REM
2560 FOR I=N-1 TO 0 STEP -1
2570 P(1)=B(1)+P(I+1)*XX
2580 NEXT I
2590 N=N-1
2600 FOR I=0 TO N
2610 P(1)=B(1+I)
2620 NEXT I
2630 RETURN
2640 PRINT "VUOI CONTINUARE" (S/N)*
2650 CALL KEY(0,K,S)
2660 IF S=0 THEN 2650
2670 IF I=87 THEN 410
2680 IF I=115 THEN 410
2690 IF I=78 THEN 2720
2700 IF I=110 THEN 2720
2710 GOTO 2650
2720 CALL CLEAR
2730 END

```

Il listato contiene alcune REM che rendono più agevole la sua lettura e che rendono superflue ulteriori precisazioni. Nella parte iniziale sono definiti i caratteri speciali e si stampa una breve presentazione. Il grado dell'equazione viene posto in G e successivamente trasferito in N, in questo modo N potrà variare senza perdere il valore iniziale G. I coefficienti vengono inseriti nel vettore A(I). Le righe 720-870 scrivono nel modo convenzionale l'equazione prima di cercare le soluzioni; l'utente può controllare di aver inserito i dati in modo corretto ed ha la conferma che le soluzioni si riferiscono proprio alla sua equazione. Alle linee 880-900 si distingue se l'equazione è di primo, secondo grado o di grado maggiore poiché si seguono tre algoritmi diversi. Il primo di questi occupa le sole righe 910-930, il secondo le righe 950-1470, il terzo le righe 1480-2080.

La variabile CTR (1480) è un contatore che indica il numero di soluzioni trovate; queste vengono memorizzate nel vettore R(CTR). Poiché, come è stato detto, ogni volta che viene trovata una soluzione si può individuare una equazione di grado immediatamente inferiore, è necessario utilizzare un altro vettore B(I) per porvi i coefficienti.

Durante l'elaborazione si opererà sempre sul vettore B(I).

Alla linea 1630, X0 rappresenta il punto dal quale inizia la ricerca di una radice dell'equazione con il metodo di Newton; convenzionalmente è stato posto $X0=0$. Con la routine 2150 si calcola il valore P del polinomio al primo membro dell'equazione e della sua derivata P1 nel punto X0. Da questi si ricava (linea 2370) $X=X0-P/P1$ e, dopo aver posto $X0=X$,

si ripete il ciclo. Questo termina in uno dei seguenti casi: (1) si sono effettuate più di 50 iterazioni, infatti generalmente ne bastano una decina; (2) il polinomio si azzerava in X_0 , nel qual caso è $P=0$; (3) la derivata si annulla in X_0 . Queste condizioni vengono valutate alle linee 1650-1670. Se il programma non riesce a trovare una soluzione viene scelto casualmente un altro punto dal quale iniziare nuovamente la ricerca e, se l'esito è ancora negativo, si chiede all'utente di esprimere un valore dal quale iniziare. Questo numero dovrà essere prossimo ad una presunta radice dell'equazione.

Il programma occupa 5 Kbyte di memoria.

Come test si può porre uguale a 3 il grado dell'equazione e scrivere i seguenti coefficienti: 1, 6, 3, -10; dopo un paio di secondi appariranno le soluzioni: 1, -5, -2.

La scritta VUOI CONTINUARE? (S/N) indica che l'elaborazione è finita. Se per un'equazione di terzo grado dovesse apparire una sola soluzione è evidente che le altre radici non sono reali.

Da un punto di vista didattico può essere interessante vedere le successive approssimazioni di una radice calcolate con il metodo di Newton. A questo proposito la REM di linea 2380 ricorda che in quella stessa posizione si può inserire una istruzione per stampare X_0 e P . Si noterà allora che i due valori fluttueranno e, quando sarà $P=0$, X_0 è una radice reale dell'equazione.

Determinanti e soluzione di sistemi lineari

Non a caso i due argomenti sono stati associati, infatti una soluzione di sistemi lineari viene eseguita ricorrendo alla regola di Cramer che prevede l'uso dei determinanti.

Per evitare inutili ripetizioni, è buona norma utilizzare una stessa routine per svolgere funzioni diverse.

Un determinante di secondo ordine si indica con il simbolo:

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

ed è uguale al numero $a_{11} \times a_{22} - a_{21} \times a_{12}$.

Gli indici della lettera a si riferiscono il primo alla riga, il secondo alla colonna.

Dato un determinante di terzo ordine, si dice complemento algebrico di un suo elemento il determinante di secondo ordine che si ottiene dal primo eliminando la riga e la colonna relative all'elemento stesso, preceduto dal segno più o meno, a seconda che sia pari o dispari la somma degli indici.

Il determinante è uguale alla somma degli elementi di una riga per i rispettivi complementi algebrici.

La medesima definizione può essere estesa a determinanti di ordine superiore.

Il programma permette di risolvere determinanti di qualsivoglia ordine n , con $n \leq 10$; nulla vieta, tuttavia, di dimensionarlo diversamente, aggiungendo l'istruzione DIM (per $n \leq 10$ l'istruzione non è necessaria) e modificando le linee di protezione dopo l'input del numero N .

La tecnica di programmazione è originale.

Sia N l'ordine del determinante, si definisce $K=N-2$, e, utilizzando la stessa routine del programma sulle disposizioni (vedi capitolo 5), si creano le disposizioni semplici con gli N indici di colonna, presi a K a K . Gli indici di colonna, oggetti di ogni disposizione, vengono associati agli indici di riga 1, 2, 3 ... K , vengono quindi moltiplicati tra loro gli elementi del determinante con quegli indici di riga e colonna; tale prodotto viene ancora moltiplicato per il determinante del secondo ordine che si ottiene con gli elementi delle ultime due righe, $N-1$ ed N , ed aventi numeri di colonna diversi da quelli che compaiono come oggetti della disposizione semplice. Il prodotto degli elementi associati agli oggetti di ogni disposizione semplice e del determinante del secondo ordine, costituisce un addendo che concorre al calcolo del determinante; tali addendi vengono presi alternativamente con il segno più e meno; il loro numero è uguale al numero delle disposizioni semplici che si ottengono con N oggetti presi a K a K .

Un esempio pratico chiarirà meglio l'algoritmo seguito. Sia dato il seguente determinante del quarto ordine:

$$\begin{vmatrix} 2 & 3 & -1 & 4 \\ -1 & 2 & 3 & 5 \\ 4 & -2 & 1 & 3 \\ 1 & 1 & 2 & -4 \end{vmatrix}$$

È $N=4$ e $K=N-2=2$; il numero delle disposizioni semplici è $D_{4,2} = 4 \times 3 = 12$ e, prendendo come oggetti gli indici di colonna, esse sono: 1,2; 1,3; 2,1; 2,3; 2,4; ecc.

Di ogni coppia, il primo indice di colonna viene associato al numero di riga 1 ed il secondo al numero di riga 2. Data la prima disposizione (1,2) vengono individuati gli elementi $a_{11}=2$ e $a_{22}=2$ e successivamente il determinante di secondo ordine formato dalle ultime due righe e dalle colonne con indice diverso da 1 e 2, cioè 3 e 4; esso è $\begin{vmatrix} 1 & 3 \\ 2 & -4 \end{vmatrix}$ e viene

calcolato nel modo consueto dando il risultato -10 .

Il primo addendo sarà pertanto: $2 \times 2 \times (-10) = -40$.

La seconda disposizione è (1,3), corrispondente agli elementi del deter-

minante $a_{11}=2$ e $a_{22}=3$; il determinante del secondo ordine è formato dalle ultime due righe e dalle colonne con indice diverso da 1 e 3, cioè 2 e 4. Esso è

$$\begin{vmatrix} -2 & 3 \\ 1 & -4 \end{vmatrix} = 5$$

Il secondo addendo, ricordando che essi vengono presi alternativamente col segno più e meno, è: $-2 \times 3 \times 5 = -30$.

Ripetendo il procedimento per le 12 disposizioni, si ottengono 12 prodotti la somma dei quali dà il valore del determinante. Il tempo di elaborazione è di circa un secondo per ogni ciclo; di conseguenza un determinante di quarto ordine richiede per il calcolo circa dodici secondi.

I sistemi lineari sono risolti ricorrendo alla regola di Cramer. I coefficienti delle n incognite formano un determinante di ordine n che viene detto determinante del sistema, il quale si suppone sia diverso da zero. Sostituendo i termini noti, rispettivamente, ai coefficienti di ogni incognita, si ottengono i determinanti $D_1, D_2, \dots, D_n$.

La soluzione del sistema è:

$$x_1 = \frac{D_1}{D}, \quad x_2 = \frac{D_2}{D}, \quad x_n = \frac{D_n}{D}$$

Il programma, utilizzando la routine per la soluzione dei determinanti, fornisce la soluzione del sistema in forma frazionaria, dove i termini delle frazioni sono proprio i valori dei determinanti sopra definiti; eventualmente le stesse frazioni sono ridotte ai minimi termini.

```

100 REM      DETERMINANTI
110 REM      E SOLUZIONI
120 REM      DI SISTEMI LINEARI
130 REM      *****
140 REM      DI: SERGIO BORSANI
150 REM      VERSIONE: TI BASIC
160 REM
170 CALL CLEAR
180 CALL CHAR(128,"F")
190 CALL COLOR(13,9,9)
200 PRINT TAB(9):"DETERMINANTI":
210 PRINT TAB(9):"E SOLUZIONI":
220 PRINT TAB(6):"DI SISTEMI LINEARI":
230 CALL MCHAR(10,6,128,22)
240 CALL VCHAR(11,6,128,7)
250 CALL VCHAR(11,27,128,7)
260 CALL MCHAR(18,6,128,22)
270 FOR T=1 TO 300
280 NEXT T
290 CALL CHAR(136,"00000000FF")
300 CALL CHAR(137,"301010101")
310 CALL CHAR(138,"7808784078")
320 CALL CHAR(139,"5848484848")
330 CALL CHAR(140,"0408080810101010")
340 CALL CHAR(141,"3030303030303030")
350 CALL CHAR(142,"1010101010202040")
360 CALL CHAR(143,"2020101010101030")
370 CALL CHAR(144,"1010101008080804")
380 CALL CLEAR

```

```

390 PRINT "PREMI:":
400 PRINT "1. PER CALCOLARE IL VALORE": DI UN DETERMINANTE":
410 PRINT "2. PER RISOLVERE UN SISTEMA": LINEARE":
420 PRINT "3. PER FINIRE.":
430 CALL KEY(0,KK,SS)
440 IF SS=0 THEN 430
450 IF KK=49 THEN 490
460 IF KK=50 THEN 1440
470 IF KK=51 THEN 3100
480 GOTO 430
490 CALL CLEAR
500 FLAG=0
510 PRINT "DETERMINANTI":
520 CALL MCHAR(24,3,128,28)
530 PRINT ":
540 PRINT "SCRIVI L'ORDINE DEL DETERMINANTE, (N<=10).":
550 INPUT "N = ":N
560 PRINT
570 IF N=1 THEN 600
580 PRINT "ERRORE! N NON PUO' ESSERE": "MINORE DI 1, RIPETI":
590 GOTO 550
600 IF N<=10 THEN 630
610 PRINT "SPIACENTE, N NON PUO' ESSERE": "MAGGIORE DI 10, RIPETI":
620 GOTO 550
630 IF FLAG=1 THEN 1700
640 PRINT "SCRIVI, RIGA PER RIGA, GLI ELEMENTI DEL DETERMINANTE":
650 FOR I=1 TO N
660 PRINT "RIGA":I":
670 FOR J=1 TO N
680 PRINT "COLONNA":J": " = "
690 INPUT A(I,J)
700 NEXT J
710 PRINT
720 NEXT I
730 IF N=2 THEN 1390
740 IF N=1 THEN 1350
750 GOSUB 770
760 GOTO 1270
770 DTM=0
780 K=N-2
790 NI=N-1
800 D=N
810 FOR I=1 TO K-1
820 D=D*(N-I)
830 NEXT I
840 PRINT :
850 FOR I=3 TO N
860 CTR(I)=1
870 P(I)=N+1-I
880 NEXT I
890 S=1
900 FOR CNT=1 TO D
910 F=1
920 FOR R=1 TO K
930 C=P(N+1-R)
940 F=F*A(R,C)
950 NEXT R
960 CD=1
970 FOR I=1 TO N
980 FOR J=3 TO N
990 IF P(J)=I THEN 1030
1000 NEXT J
1010 B(CD)=1
1020 CD=2
1030 NEXT I
1040 D2=A(N1,B(1))*A(N,B(2))-A(N,B(1))*A(N1,B(2))
1050 F=F*D2*S
1060 DTM=DTM+F
1070 S=-S
1080 IF CNT=D THEN 1260
1090 K=J
1100 CTR(X)=CTR(X)+1
1110 IF CTR(X)<=X THEN 1160
1120 CTR(X)=1
1130 X=X+1
1140 IF X>N THEN 1260
1150 GOTO 1100
1160 FOR W=X TO 3 STEP -1
1170 P(W)=P(W)+1
1180 IF W=N THEN 1240
1190 IF P(W)<=N THEN 1210
1200 P(W)=1
1210 FOR I=W+1 TO N
1220 IF P(I)=P(W) THEN 1170
1230 NEXT I
1240 NEXT W
1250 NEXT CNT

```

```

1260 RETURN
1270 CALL CLEAR
1280 PRINT "IL VALORE DEL DETERMINANTE": "E'":
1290 PRINT "D =" ; DTM
1300 PRINT :
1310 PRINT "VUOI CONTINUARE? (S/N)"
1320 CALL KEY(0, KK, 99)
1330 IF 99=0 THEN 1320
1340 IF KK=83 THEN 490 ELSE 380
1350 PRINT :
1360 PRINT "IL VALORE DEL DETERMINANTE": "E'":
1370 PRINT "D =" ; A(1,1)
1380 GOTO 1310
1390 PRINT :
1400 DTH=A(1,1)*A(2,2)-A(2,1)*A(1,2)
1410 PRINT "IL VALORE DEL DETERMINANTE": "E'":
1420 PRINT "D =" ; DTH
1430 GOTO 1310
1440 CALL CLEAR
1450 FLAG=1
1460 PRINT "SOLUZIONE": "DI UN SISTEMA LINEARE":
1470 CALL HCHAR(24,3,128,28)
1480 PRINT :
1490 PRINT "UN SISTEMA LINEARE PUO' ES-": "SERE RAPPRESENTATO NEL MODO"
1500 PRINT "SEGUENTE":
1510 PRINT CHR$(140);
1520 GOSUB 2480
1530 PRINT CHR$(141); " "; CHR$(137); CHR$(137); " "; CHR$(137);
1540 PRINT " "; CHR$(137); CHR$(138); " "; CHR$(138);
1550 PRINT TAB(19); CHR$(137); CHR$(139); " "; CHR$(139); " "; CHR$(137);
1560 PRINT CHR$(141);
1570 GOSUB 2480
1580 PRINT CHR$(142); " "; CHR$(138); CHR$(137); " "; CHR$(137); " ";
1590 PRINT CHR$(138); CHR$(138); " "; CHR$(138); TAB(19); CHR$(138); CHR$(139);
1600 PRINT " "; CHR$(139); " "; CHR$(138);
1610 PRINT CHR$(143); " . . . . . "
1620 PRINT CHR$(141);
1630 PRINT CHR$(141);
1640 GOSUB 2480
1650 PRINT CHR$(144); " "; CHR$(139); CHR$(137); " "; CHR$(137); " ";
1660 PRINT CHR$(139); CHR$(138); " "; CHR$(138);
1670 PRINT TAB(19); CHR$(139); CHR$(139); " "; CHR$(139); " "; CHR$(139);
1680 PRINT :
1690 GOTO 530
1700 PRINT "SCRIVI, NELL'ORDINE, I COEFFICIENTI ED I TERMINI NOTI":
1710 FOR I=1 TO N
1720 FOR J=1 TO N
1730 PRINT "A(" ; STR$(I) ; STR$(J) ; ")=" ;
1740 INPUT A(I,J)
1750 NEXT J
1760 PRINT "H(" ; STR$(I) ; ")=" ;
1770 INPUT H(I)
1780 PRINT
1790 NEXT I
1800 PRINT
1810 IF N<1 THEN 1930
1820 IF A(1,1)<>0 THEN 1850
1830 PRINT "L'EQUAZIONE NON E' DETERMI-": "NATA"
1840 GOTO 2400
1850 NU=H(1)
1860 DENO=A(1,1)
1870 GOSUB 2570
1880 DDH=H(1)
1890 DDS=A(1,1)
1900 G=1
1910 GOSUB 2750
1920 GOTO 2400
1930 IF N<2 THEN 2150
1940 DDS=A(1,1)*A(2,2)-A(2,1)*A(1,2)
1950 IF DDS<0 THEN 1980
1960 PRINT "IL SISTEMA NON E' DETERMINA-TO"
1970 GOTO 2400
1980 DDH=H(1)*A(2,2)-H(2)*A(1,2)
1990 NU=DDH
2000 DENO=DDS
2010 GOSUB 2570
2020 GOSUB 2500
2030 PRINT
2040 B=1
2050 GOSUB 2750
2060 DDH=A(1,1)*H(2)-A(2,1)*H(1)
2070 NU=DDH
2080 DENO=DDS
2090 GOSUB 2570
2100 GOSUB 2500
2110 PRINT

```

```

2120 G=2
2130 GOSUB 2750
2140 GOTO 2400
2150 PRINT
2160 FOR I=1 TO N
2170 FOR J=1 TO N
2180 COEF(I,J)=A(I,J)
2190 NEXT J
2200 NEXT I
2210 GOSUB 770
2220 IF DTM<>0 THEN 2240
2230 GOTO 1960
2240 DDS=DTM
2250 FOR G=1 TO N
2260 FOR J=1 TO N
2270 A(J,G)=H(J)
2280 NEXT J
2290 GOSUB 770
2300 DDX=DTM
2310 NUME=DDX
2320 DENO=DDS
2330 GOSUB 2370
2340 GOSUB 2300
2350 GOSUB 2750
2360 FOR J=1 TO N
2370 A(J,G)=COEF(J,B)
2380 NEXT J
2390 NEXT G
2400 PRINT "VUOI CONTINUARE? (S/N)"
2410 CALL KEY(0,KK,SS)
2420 IF SS=0 THEN 2410
2430 IF KK=83 THEN 1440 ELSE 380
2440 PRINT "VUOI CONTINUARE? (S/N)"
2450 CALL KEY(0,KK,SS)
2460 IF SS=0 THEN 2450
2470 IF KK=83 THEN 1440 ELSE 380
2480 PRINT "A X +A X +...+A X = H"
2490 RETURN
2500 IF DENO<>0 THEN 2530
2510 NUME=-NUME
2520 DENO=-DENO
2530 RETURN
2540 REM      SEMPLIFICAZIONE
2550 REM
2560 REM
2570 IF NUME>DENO THEN 2610
2580 Y=DENO
2590 X=NUME
2600 GOTO 2630
2610 Y=NUME
2620 X=DENO
2630 Q=Y/X
2640 R=Y-X*INT(Q)
2650 IF R=0 THEN 2690
2660 Y=X
2670 X=R
2680 GOTO 2630
2690 NUME=NUME/X
2700 DENO=DENO/X
2710 RETURN
2720 REM
2730 REM      STAMPA
2740 REM
2750 IF ABS(DDX)>ABS(DDS) THEN 2780
2760 L1=LEN(STR$(DDS))
2770 GOTO 2790
2780 L1=LEN(STR$(DDX))
2790 IF ABS(NUME)>DENO THEN 2820
2800 L2=LEN(STR$(DENO))
2810 GOTO 2830
2820 L2=LEN(STR$(NUME))
2830 PRINT TAB(6+L1-LEN(STR$(DDX)));STR$(DDX);
2840 IF DDX=NUME THEN 2880
2850 IF DENO=1 THEN 2880
2860 PRINT TAB(9+L1+L2-LEN(STR$(NUME)));STR$(NUME)
2870 GOTO 2890
2880 PRINT
2890 PRINT "X";STR$(G);" = "
2900 FOR J=6 TO 5+L1
2910 PRINT TAB(J);CHR$(136);
2920 NEXT J
2930 IF DENO<>1 THEN 2960
2940 PRINT " = ";NUME
2950 GOTO 3030
2960 IF DDX=NUME THEN 3030
2970 PRINT " = ";

```

```

2980 INIT=9+L1
2990 FINE=8+L1+L2
3000 FOR J=INIT TO FINE
3010 PRINT TAB(J);CHR$(136);
3020 NEXT J
3030 PRINT TAB(6+L1-LEN(STR$(DDS)));STR$(DDS);
3040 IF DENO=1 THEN 3080
3050 IF DENO=DDS THEN 3080
3060 PRINT TAB(9+L1+L2-LEN(STR$(DENO)));STR$(DENO)
3070 GOTO 3090
3080 PRINT
3090 RETURN
3100 CALL CLEAR
3110 END

```

Il programma è diviso in due sezioni principali, come risulta dal menù (linee 390-420). Nella prima parte, alle linee 540-720 si accettano i dati in input; N è l'ordine del determinante mentre i suoi elementi sono contenuti nella matrice A(I,J). Alle righe 730-750 si indirizza l'elaborazione all'algoritmo principale solo se N è maggiore di due, poiché altrimenti il calcolo è banale e si svolge in altra sede. Il calcolo di un determinante di ordine superiore a due è organizzato in forma di routine perché servirà anche alla soluzione dei sistemi lineari ed occupa la parte compresa tra le righe 770-1260.

All'interno di questa routine, le variabili CTR(I), P(I) e P(W) sono da mettere in relazione con la formazione delle disposizioni ed il loro significato verrà pienamente chiarito in seguito nel programma sulle disposizioni. Per quanto riguarda le altre: D è il numero delle disposizioni, F è il prodotto tra gli elementi delle colonne permutate ed il determinante di secondo ordine loro complemento algebrico, DTM è il valore del determinante, D2 è il determinante del secondo ordine, infine, S (linea 1070) inverte il segno ad ogni addendo, poiché, come è stato detto, gli addendi che concorrono a formare il determinante vanno presi alternativamente con il segno + e -.

La seconda sezione si apre con una bella rappresentazione di un sistema lineare che facilita la comprensione di come debbano essere inseriti i dati; la sua presenza non è indispensabile.

I sistemi con due equazioni vengono risolti come un caso particolare (linee 1940-2140); quelli formati da più di due equazioni vengono risolti immediatamente dopo (linee 2150-2390). DDS è il determinante del sistema, il quale deve essere diverso da 0, DDX è quello che si ottiene dal precedente sostituendo ad una variabile i termini noti (2260-2280). In entrambi i casi il determinante è calcolato dalla routine 770 già considerata in precedenza.

Il risultato è dato in forma frazionaria: $X = DDX/DDS$. I termini della frazione sono semplificati nella routine 2570 la quale altro non è che l'algoritmo euclideo.

NUME è il numeratore, DENO il denominatore. Se DENO è negativo si invertono i segni di entrambi in modo da avere un eventuale segno meno solo al numeratore.

La memoria utilizzata è di 6,4 K.

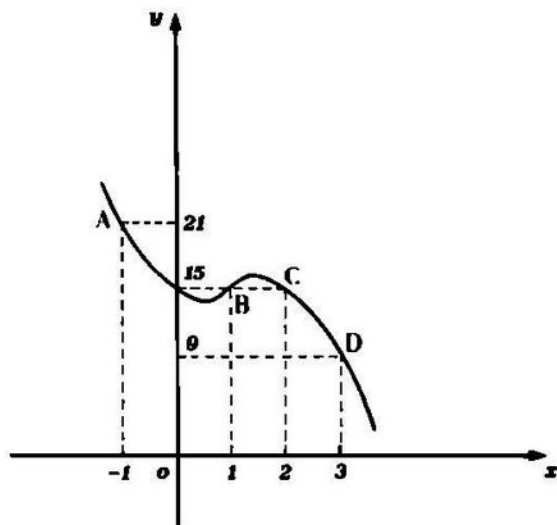


Fig. 12

Passiamo ora ad una applicazione. Si voglia risolvere il seguente problema: trovare l'equazione di terzo grado della curva passante per i punti $A(-1,21)$, $B(1,15)$, $C(2,15)$ e $D(3,9)$. La curva è illustrata nella figura 12 e la sua equazione generica è: $ax^3 + bx^2 + cx + d = y$. Il problema consiste nel trovare i coefficienti della variabile x .

Poiché la curva passa per il punto A , sostituendo le coordinate alle variabili x e y nell'equazione, deve essere soddisfatta la seguente uguaglianza

$$a(-1)^3 + b(-1)^2 + c(-1) + d = 21$$

cioè

$$-a + b - c + d = 21$$

Poiché la curva passa anche per gli altri tre punti è lecito ripetere l'operazione con le loro coordinate. Si ottengono in questo modo quattro equazioni lineari che formano un sistema.

$$\begin{cases} -a + b - c + d = 21 \\ a + b + c + d = 15 \\ 8a + 4b + 2c + d = 15 \\ 27a + 9b + 3c + d = 9 \end{cases}$$

È il momento ora di introdurre i dati nel computer.

RUN

```

N = 4                                <ENTER>
SCRIVI, NELL'ORDINE, I COEF-
FICIENTI ED I TERMINI NOTI
A(11)=-1                             <ENTER>
A(12)=1                               <ENTER>
A(13)=-1                             <ENTER>
A(14)=1                               <ENTER>
H(1)=21                              <ENTER>
A(21)=1                               <ENTER>
A(22)=1                               <ENTER>
A(23)=1                               <ENTER>
A(24)=1                               <ENTER>
H(2)=15                              <ENTER>
A(31)=8                               <ENTER>
A(32)=4                               <ENTER>
A(33)=2                               <ENTER>
A(34)=1                               <ENTER>
H(3)=15                              <ENTER>
A(41)=27                             <ENTER>
A(42)=9                               <ENTER>
A(43)=3                               <ENTER>
A(44)=1                               <ENTER>
H(4)=9                               <ENTER>

X1=  $\frac{-48}{48}$  = -1
X2=  $\frac{144}{48}$  = 3
X3=  $\frac{-96}{48}$  = -2
X4=  $\frac{720}{48}$  = 15

. VUOI CONTINUARE? (S/N)

```

I risultati rappresentano i valori delle incognite: a , b , c , d . L'equazione cercata è pertanto:

$$y = -x^3 + 3x^2 - 2x + 15$$

Integrali definiti

Data una funzione $f(x)$, continua e non negativa in un intervallo (a, b) , da un punto di vista geometrico l'integrale definito rappresenta l'area

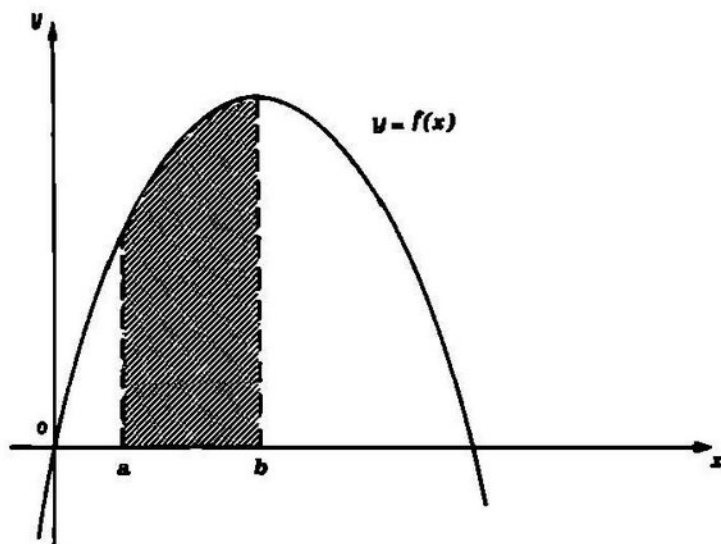


Fig. 13

del trapezoido $ABMN$ (vedi figura 13), delimitato dalla curva di equazione $y=f(x)$, dall'asse delle x e dalle parallele all'asse delle y , AM e BN . Se suddividiamo l'intervallo (a, b) in n parti di ampiezza $h = (b-a)/n$, possiamo definire la seguente somma:

$$S = f(x_1) \times h + f(x_2) \times h + \dots + f(x_n) \times h$$

dove $x_i = a + ih$, $i = 1, 2, 3, \dots, n$.

Il valore di S dipende dall'ampiezza delle parti h e pertanto dal numero delle suddivisioni n . Al variare di n avremo:

$$S_1, S_2, S_3, \dots, S_n, \dots$$

L'area del trapezoido e l'integrale definito dalla funzione $f(x)$ sono uguali al limite della successione S_n per n tendente all'infinito.

Il programma adotta tre metodi di integrazione: quello di Simpson con $2n$ intervalli, quello di Gauss con 12 punti ed un caso particolare di quest'ultimo quando l'estremo superiore dell'intervallo sia infinito.

Nel primo caso si utilizza la formula:

$$\begin{aligned} \int_a^b f(x) dx = & h/3 [f(a) + 4 [f(x_1) + f(x_3) + \dots + f(x_{2n-1})] + \\ & + 2 [f(x_2) + f(x_4) + \dots + f(x_{2n-2})] + f(b)] \end{aligned}$$

dove $h = \frac{b-a}{2n}$ e $x_i = a + ih$, con $i=1, 2, 3, \dots, 2n$.

Nel secondo caso si utilizza la seconda formula:

$$\int_a^b f(x)dx = (b-a)/2 \times \sum_{i=0}^n w_i \times f(y_i)$$

dove $y_i = v_i(b-a)/2 + (b-a)/2$. I numeri v_i e w_i sono definiti nella prima parte del programma e memorizzati nelle variabili con indici V(I) e W(I).

Nel terzo caso si usa la formula:

$$\int_a^b f(x)dx = 2 \times \sum_{i=0}^n w_i(1+v_i)^2 f(y_i)$$

dove $y_i = 2/(1+v_i) + a - 1$. I numeri v_i e w_i sono gli stessi visti in precedenza.

Come piccola concessione alla forma si è voluto definire il simbolo di integrale che appare quando vengono dati i risultati.

La funzione integranda deve essere scritta nell'istruzione DEF di linea 200 seguendo le istruzioni che appaiono sullo schermo; dopo di che, non consentendo il TI Basic il comando RUN 200, il programma deve riprendere dall'inizio.

Successivamente bisogna inserire gli estremi dell'intervallo nel quale si vuole integrare la funzione e, solo se si adotta il metodo di Simpson, va specificato in quante parti suddividere l'intervallo $b-a$. Aumentando n aumenta la precisione, tuttavia è sufficiente porre $n=10$ se $b-a$ ha un valore unitario.

La memoria utilizzata è di 3,3K.

La tecnica di programmazione non presenta difficoltà e consiste nel seguire le formule date.

Ecco il listato.

```

100 REM      INTEGRALI
110 REM      *****
120 REM      di Sergio Borsani
130 REM      versione: TI BASIC
140 CALL CLEAR
150 DIM V(11), W(11)
160 PRINT TAB(5); "INTEGRALI DEFINITI":;
170 FOR I=1 TO 600
180 NEXT I
190 CALL CLEAR
200 DEF F(X)=-X*X+4*X
210 PRINT "      DEFINIRE LA FUNZIONE DA"
220 PRINT "INTEGRARE ALLA LINEA 200 DEL"
230 PRINT "PROGRAMMA, SE SI E' GIA'"
240 PRINT "PROVVEDUTO PREMERE UN TASTO"
250 PRINT "PER CONTINUARE":;
260 CALL KEY(0, F, S)
270 IF S=0 THEN 260
280 CALL CHAR(128, "0000070808081818")
290 CALL CHAR(129, "18181818181818")
300 CALL CHAR(130, "18181818181818")
305 CALL CHAR(131, "000000001D22221D")

```

```

310 CALL CHAR(132,"00000000C02020C")
320 FF=0
330 CALL CLEAR
340 ES=0
350 PRINT "SCEGLI TRA I SEGUENTI METODI"
360 PRINT "DI INTEGRAZIONE":
370 PRINT "1. SIMPSON":
380 PRINT "2. GAUSS":
390 PRINT "3. GAUSS CON ESTREMO SUPE-      RIORE INFINITO":
400 INPUT "SCELTA? ":S
410 IF (S>0)*(S<4) THEN 450
420 PRINT
430 PRINT "DEVI PREMERE 1,2 O 3":
440 GOTO 400
450 ON S GOTO 460,700,820
460 CALL CLEAR
470 GOSUB 1400
480 PRINT
490 PRINT "L'INTERVALLO VERRA' SUDDIVI-SO IN 2N PARTI":
500 INPUT "N = ":N
510 PRINT :
520 INTEG=0
530 H=(X2-X1)/(2*N)
540 SOMMA=0
550 N1=2*N-1
560 FOR I=1 TO N1 STEP 2
570 X=X1+I*H
580 SOMMA=SOMMA+F(X)
590 NEXT I
600 INTEG=SOMMA*4
610 SOMMA=0
620 N2=N1-1
630 FOR I=2 TO N2 STEP 2
640 X=X1+I*H
650 SOMMA=SOMMA+F(X)
660 NEXT I
670 INTEG=INTEG+2*SOMMA+F(X1)+F(X2)
680 INTEG=INTEG*H/3
690 GOTO 980
700 CALL CLEAR
710 GOSUB 1400
720 INTEG=0
730 IF FF=1 THEN 750
740 GOSUB 1110
750 FOR I=0 TO 11
760 Y=(X2-X1)*V(I)/2
770 Y=Y+(X2-X1)/2
780 INTEG=INTEG+W(I)*F(Y)
790 NEXT I
800 INTEG=INTEG*(X2-X1)/2
810 GOTO 980
820 CALL CLEAR
830 PRINT "INTERVALLO DI INTEGRAZIONE"
840 PRINT "DA X1 ALL'INFINITO":
850 INPUT "X1 = ":X1
860 PRINT
870 IF FF=1 THEN 890
880 GOSUB 1110
890 INTEG=0
900 FOR I=1 TO 11
910 V=(2/(1+V(I)))*X1-1
920 Y1=F(Y)/(1+V(I)^2)
930 INTEG=INTEG+W(I)*Y1
940 NEXT I
950 INTEG=INTEG*2
960 ES=1
980 PRINT :
990 IF ES=0 THEN 1020
1000 PRINT TAB(3);CHR$(131);CHR$(132)
1010 GOTO 1030
1020 PRINT TAB(2);X2
1030 PRINT TAB(3);CHR$(128)
1040 PRINT TAB(3);CHR$(129);"F(X)DX =" ;INTEG
1050 PRINT TAB(3);CHR$(130)
1060 PRINT TAB(2);X1:
1070 GOTO 1520
1080 REM
1090 REM      X(I),W(I)
1100 REM
1110 V(0)=-.1252334085
1120 V(1)=-.3678314989
1130 V(2)=-.5873179342
1140 V(3)=-.7699026741
1150 V(4)=-.9041172563
1160 V(5)=-.9813606342
1170 V(6)=-V(0)

```

```

1180 V(7)=-V(1)
1190 V(8)=-V(2)
1200 V(9)=-V(3)
1210 V(10)=-V(4)
1220 V(11)=-V(5)
1230 W(0)=.2491470458
1240 W(1)=.2334925365
1250 W(2)=.2031674267
1260 W(3)=.1600783285
1270 W(4)=.1069373259
1280 W(5)=.0471753363
1290 W(6)=W(0)
1300 W(7)=W(1)
1310 W(8)=W(2)
1320 W(9)=W(3)
1330 W(10)=W(4)
1340 W(11)=W(5)
1350 FF=1
1360 RETURN
1370 REM
1380 REM     INTERVALLO
1390 REM
1400 PRINT "SCRIVI GLI ESTREMI DELL'IN-"
1410 PRINT "TERVALLO DI INTEGRAZIONE":
1420 PRINT "ESTREMO INFERIORE"
1430 INPUT "X1 = ":X1
1440 PRINT
1450 PRINT "ESTREMO SUPERIORE"
1460 INPUT "X2 = ":X2
1470 IF X1<X2 THEN 1510
1480 PRINT
1490 PRINT "DEVE ESSERE X1<X2, RIPROVA!":
1500 GOTO 1420
1510 RETURN
1520 PRINT ":
1530 PRINT "VUOI CONTINUARE? (S/N) "
1540 CALL KEY(0,K,S)
1550 IF S=0 THEN 1540
1560 IF (K=83)+(K=115) THEN 1330
1570 IF (K=78)+(K=110) THEN 1390
1580 GOTO 1540
1590 CALL CLEAR
1600 END

```

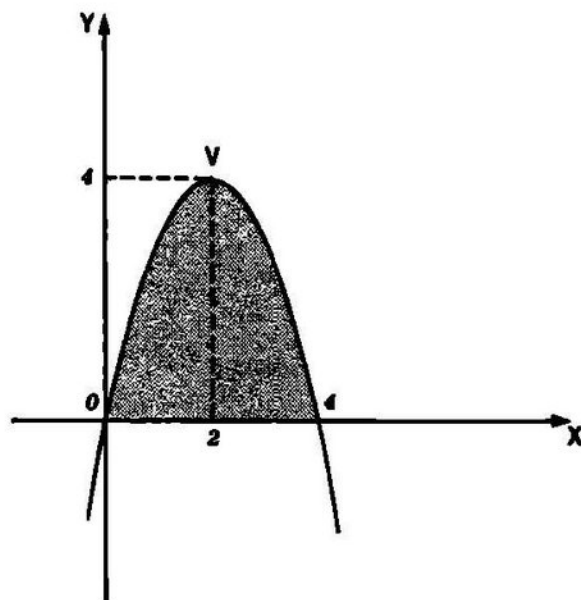


Fig. 14

Vediamo ora qualche applicazione.

1. Si voglia calcolare l'area della superficie limitata dalla parabola di equazione $y = -x^2 + 4x$ e l'asse delle x .

La parabola incontra l'asse delle x nei punti di ascissa 0 e 4, pertanto sono questi gli estremi dell'intervallo di integrazione (vedi figura 14).

L'area cercata è:

$$\text{Area} = \int_0^4 f(x)dx = \int_0^4 (-x^2 + 4x)dx$$

Definiamo la funzione alla linea 200 del programma, otterremo il seguente output.

RUN

```
200                                <FCTN(<>)>
200 DEF F(X)=-2*X*X+4*X          <ENTER>
```

```
SCELTA? 1                        <ENTER>
```

```
SCRIVI GLI ESTREMI DELL'INTER-  
VALLO DI INTEGRAZIONE
```

```
ESTREMO INFERIORE
```

```
X1 = 0
```

```
<ENTER>
```

```
ESTREMO SUPERIORE
```

```
X2 = 2
```

```
<ENTER>
```

```
L'INTERVALLO VERRA' SUDDIVI-  
SO IN 2N PARTI
```

```
N = 10
```

```
<ENTER>
```

```
2
```

```
 $\int_0^2 F(X)DX = 2.666666667$ 
```

```
0
```

```
VUOI CONTINUARE? (S/N)
```

Con carta e matita il risultato sarebbe stato $8/3$, quindi in perfetto accordo con il valore decimale ottenuto al computer.

Archimede seppe elaborare un metodo per dare una risposta a questo tipo di problemi più di 2000 anni fa!

2. Si voglia determinare la lunghezza di un arco della parabola di equazione $y = x^2$ avente per estremi i punti $O(0,0)$ e $P(1,1)$ (vedi figura 15).

La formula per trovare la lunghezza di un arco di curva piana è:

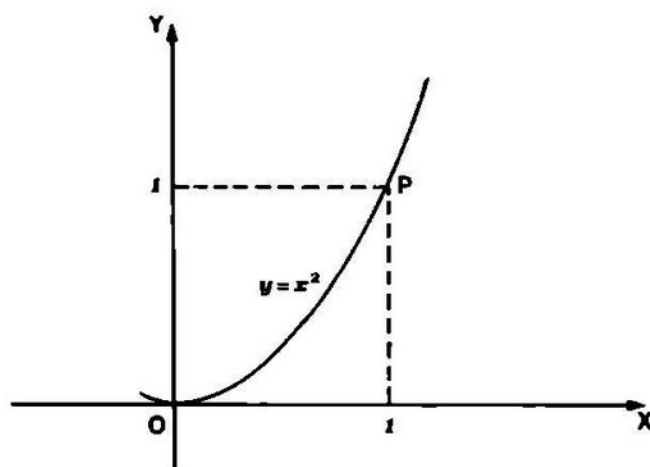


Fig. 16

$$L = \int_a^b \sqrt{1+f'^2(x)} dx$$

Poiché $f(x)=x^2$, la sua derivata è: $f'(x)=2x$ e $f'^2(x)=4x^2$.

Alla linea 200 del programma definiamo la seguente funzione:

$$\text{DEF F(X)} = \text{SQR}(1+4\cdot\text{X}\cdot\text{X})$$

Avviamo nuovamente il programma ed ecco cosa apparirà sullo schermo.

RUN

```

ESTREMO INFERIORE
X1 = 0                                <ENTER>
ESTREMO SUPERIORE
X2 = 1                                <ENTER>
L'INTERVALLO VERRA' SUDDIVI-
SO IN 2N PARTI
N = 10                                <ENTER>

 $\int_0^1 F(X)DX = 1.478942828$ 

VUOI CONTINUARE? (S/N)

```

I risultati ottenuti con il metodo di Simpson ($N=10$) e di Gauss concordano fino alla settima cifra decimale.

Calcolo combinatorio

Introduzione

Secondo la definizione classica la *probabilità* che si verifichi un evento è data dal rapporto tra il numero dei casi favorevoli ed il numero di casi possibili, purché tutti i casi siano ugualmente probabili. Ad esempio, la probabilità di ottenere sei, lanciando un dado ben tarato, è $1/6$ e la probabilità di ottenere un numero pari è $3/6$, cioè $1/2$.

Talvolta tuttavia, calcolare il numero di casi possibili potrebbe presentare qualche difficoltà. Nel gioco del poker, quante differenti mani si possono formare? Qual è la probabilità di ricevere una scala massima di cuori servita?

In questi problemi non è necessario, e sarebbe praticamente impossibile, enumerare tutti i casi possibili ma ci viene in aiuto il calcolo combinatorio.

Disposizioni e permutazioni

Come la maggior parte delle routine presenti in questo libro, anche quella per creare le disposizioni semplici, utilizzata nel programma che segue, è del tutto originale e non costituisce un adattamento da altre versioni di Basic.

Dati n oggetti, si chiamano *disposizioni semplici* tutti i gruppi che si possono formare prendendoli a k a k , in modo che ogni gruppo differi-

sca dagli altri o per qualche oggetto o per l'ordine con cui gli oggetti sono disposti.

k deve essere minore o al massimo uguale ad n ; in ogni gruppo un oggetto può comparire una sola volta, cioè non sono ammesse ripetizioni. Quando $k = n$, ogni gruppo contiene tutti gli n oggetti ed i gruppi differiscono tra di loro solo per l'ordine con cui compaiono gli oggetti. Gli anagrammi sono un esempio di permutazione semplice.

Poiché le permutazioni sono un caso particolare delle disposizioni, nel programma una sola routine svolge entrambe le funzioni e, nell'esposizione, ci riferiremo in generale alle disposizioni semplici.

Il numero delle disposizioni è:

$$(1) \quad D_{n,k} = n \times (n-1) \times (n-2) \times \dots \times (n-k+2) \times (n-k+1)$$

Ad esempio, per $n=4$ e $k=3$, risulta:

$$D_{4,3} = 4 \times 3 \times 2 = 24$$

Come si vede è semplicissimo calcolare il numero delle disposizioni. Più difficile è scrivere un programma Basic in grado di scrivere tutti i gruppi di n oggetti presi a k a k con n e k variabili.

Se indichiamo con a_1, a_2, a_3, a_4 i quattro oggetti, le 24 disposizioni semplici sono:

$a_1a_2a_3$	$a_2a_1a_3$	$a_3a_1a_2$	$a_4a_1a_2$
$a_1a_2a_4$	$a_2a_1a_4$	$a_3a_1a_4$	$a_4a_1a_3$
$a_1a_3a_2$	$a_2a_3a_1$	$a_3a_2a_1$	$a_4a_2a_1$
$a_1a_3a_4$	$a_2a_3a_4$	$a_3a_2a_4$	$a_4a_2a_3$
$a_1a_4a_2$	$a_2a_4a_1$	$a_3a_4a_1$	$a_4a_3a_1$
$a_1a_4a_3$	$a_2a_4a_3$	$a_3a_4a_2$	$a_4a_3a_2$

Come si vede, per evitare ripetizioni ed avere la certezza di aver scritto in modo corretto tutti i gruppi si è seguito un certo criterio. Vediamolo attentamente perché esso è alla base del metodo adottato nella programmazione.

Stabiliamo di scrivere gli elementi di un gruppo, iniziando con gli oggetti aventi indice inferiore, pertanto al primo posto troveremo prima a_1 , poi a_2 , ecc.

È importante la seguente considerazione. Si può affermare che il primo elemento di ogni gruppo possiede quattro gradi di libertà, infatti sarà uno dei quattro oggetti.

Il secondo elemento, poiché non sono ammesse ripetizioni, possiede tre gradi di libertà, infatti, fissando l'oggetto al primo posto, al secondo sa-

rà inserito solo uno degli altri tre oggetti. Analogamente, il terzo elemento ha due gradi di libertà.

Formato il primo gruppo si prosegue cambiando l'ultimo elemento, quello con minor grado di libertà; quando ha assunto i due possibili valori è necessario variare l'elemento con grado di libertà immediatamente superiore, controllando ogni volta che non ci siano ripetizioni.

Nel programma N è il numero degli oggetti, K la classe e $J = N - K + 1$ è il minore dei fattori nella formula (1) vista in precedenza.

I contatori $CTR(X)$ hanno l'indice uguale ai gradi di libertà dell'elemento del gruppo a cui si riferiscono.

Essi vengono inizializzati a uno e riportati a questo valore quando risulta $CTR(X) > X$, cioè quando l'elemento ha assunto tutti i valori consentiti dal suo grado di libertà. Se la classe è 3, come nel nostro esempio, altrettanto sono le variabili di controllo, nell'ordine: $CTR(4)$, $CTR(3)$, $CTR(2)$. X inizialmente prende il valore più basso J ; si incrementa $CTR(2)$ finché non è maggiore di 2; poi si pone $CTR(2) = 1$ e si incrementa $CTR(3)$, finché il suo valore non è maggiore di 3; si riporta $CTR(3)$ uguale a 1 e si prosegue incrementando $CTR(4)$.

Gli elementi che formano i gruppi sono caricati nel vettore $P(I)$; anche in questo caso l'indice è uguale al grado di libertà dell'elemento corrispondente. $P(I)$ viene inizializzato con i valori 1, 2, 3. Per la stampa delle lettere dell'alfabeto basta riferirsi al numero di codice ASCII con l'introduzione $CHR(P(I) + 64)$; se $P(I) = 1$ è stampata la lettera A (codice 65), se $P(I) = 2$ è stampata la lettera B (codice 66), e così via.

Quando viene incrementato il contatore $CTR(X)$ si incrementa anche $P(X)$, con la convenzione che se risulta $P(X) > N$, $P(X)$ assume il valore più basso diverso da quello attribuito in quel momento agli elementi con grado di libertà superiore. Si controlla poi l'eventuale ripetizione di oggetti. Ad esempio, se l'ultima disposizione trovata è $a_1 a_3 a_4$ sarà $CTR(4) = 1$, $CTR(3) = 2$, $CTR(2) = 2$ e $P(4) = 1$, $P(3) = 3$, $P(2) = 4$.

Il programma prosegue incrementando $CTR(2)$; poiché il nuovo valore supera il suo stesso indice, essa viene riportata uguale a 1 e si passerà ad incrementare sia $CTR(3)$ che l'elemento con lo stesso indice $P(3)$ controllando che il nuovo valore non sia uguale a quello dell'elemento con indice immediatamente superiore. Infine si controlla che l'elemento $P(2)$ non abbia lo stesso valore di quelli con indice superiore, nel qual caso gli verrà attribuito il valore più piccolo tra quelli per così dire liberi. Al termine del ciclo i nuovi valori dei vettori sono 1, 3, 1 per $CTR(X)$ e 1, 4, 2 per $P(X)$, corrispondenti, questi ultimi, agli oggetti $a_1 a_4 a_2$.

I vettori $CTR(X)$ e $P(X)$ non sono stati dimensionati ed i loro indici devono essere pertanto non maggiori di 10. Naturalmente è possibile dimensionarli diversamente, aggiungendo l'istruzione DIM e modificando le istruzioni di protezione dopo l'input del numero degli oggetti N .

Ricordiamo tuttavia che il numero delle disposizioni aumenta veloce-

mente all'aumentare di N ed è alquanto improbabile che qualcuno senta il desiderio di trascorrere qualche ora al computer per vedere sfilare anche solo le 5040 disposizioni semplici di 10 oggetti della classe 4.

```

100 REM  DISPOSIZIONI SEMPLICI
110 REM  *****
120 REM  dir Sergio Borsani
130 REM  versione: TI BASIC
140 CALL CLEAR
150 PRINT TAB(4); "CALCOLO COMBINATORIO":::
160 FOR T=1 TO 500
170 NEXT T
180 CALL CHAR(128,"000000FF")
190 CALL CLEAR
200 PRINT "PRIMI":
210 PRINT "1. PER LE DISPOSIZIONI SEM-": PLICI DI N OGGETTI::
220 PRINT "2. PER LE PERMUTAZIONI DI N": OGGETTI::
230 PRINT "3. PER FINIRE":
240 CALL KEY(0,KK,SS)
250 IF SS=0 THEN 240
260 IF KK=49 THEN 300
270 IF KK=50 THEN 1640
280 IF KK=51 THEN 3030
290 GOTO 240
300 CALL CLEAR
310 PRINT "PRIMI":
320 PRINT "1. PER LE SPIEGAZIONI":
330 PRINT "2. PER LE APPLICAZIONI":
340 CALL KEY(0,KK,SS)
350 IF SS=0 THEN 340
360 IF KK=49 THEN 390
370 IF KK=50 THEN 890
380 GOTO 340
390 CALL CLEAR
400 PRINT "SIANO DATI QUATTRO OGGETTI":
410 PRINT TAB(10); "A B C D":
420 PRINT "PRENDENDOLI A TRE A TRE, SI"
430 PRINT "FORMINO DEI GRUPPI CHE DIFF-"
440 PRINT "FERISCANO O PER QUALCHE OG-"
450 PRINT "GETTO O PER L'ORDINE IN CUI"
460 PRINT "SONO DISPOSTI":
470 PRINT "SCRIVIAMOLI":
480 GOSUB 2940
490 CALL MCHAR(23,3,32,28)
500 PRINT ::
510 CALL MCHAR(23,3,128,28)
520 N=4
530 K=3
540 J=N-K+1
550 GOSUB 2980
560 C=0
570 C=C+1
580 IF C>24 THEN 710
590 IF C>9 THEN 610
600 PRINT " ";
610 PRINT C; "- ";
620 FOR I=0 TO K-1
630 PRINT CHR$(P(N-I)+64); " ";
640 NEXT I
650 IF C/2=INT(C/2) THEN 680
660 PRINT " ";
670 GOTO 690
680 PRINT
690 GOSUB 2770
700 GOTO 570
710 PRINT ::
720 CALL MCHAR(23,3,128,28)
730 GOSUB 2940
740 CALL CLEAR
750 PRINT "IL NUMERO DELLE DISPOSIZIONI": "E":
760 PRINT TAB(3); "D(4,3) = 4*3*2 = 24":
770 PRINT "IN GENERALE, PRESI N OGGETTI":
780 PRINT "A K A K, O DELLA CLASSE K, E":
790 PRINT "D(N,K) = N*(N-1)*...*(N-K+1)":
800 PRINT "VUOI RIVEDERE LE DISPOSIZIO-":
810 PRINT "NI DELL'ESEMPIO PRECEDENTE?": "(S/N)"
820 CALL KEY(0,KK,SS)
830 IF SS=0 THEN 820
840 IF KK=83 THEN 870
850 IF KK=78 THEN 190
860 GOTO 820
870 CALL CLEAR

```

```

880 GOTO 500
890 CALL CLEAR
900 PRINT "QUANTI SONO GLI OGGETTI?"
910 PRINT "(NON PIU' DI 10)":
920 INPUT "N =":N
930 IF N<1 THEN 960
940 PRINT "ATTENZIONE! DEVE ESSERE":N<=10, "RIPROVA.":
950 GOTO 920
960 PRINT ":
970 PRINT "QUANTI OGGETTI FORMANO OGNI": "GRUPPO?"
980 INPUT "K = ":K
990 PRINT ":
1000 D=N
1010 FOR I=1 TO K-1
1020 D=D*(N-I)
1030 NEXT I
1040 PRINT "IL NUMERO DELLE DISPOSIZIONI": "E":
1050 PRINT "D(STR(N))": "STR(K)": "D":
1060 PRINT "VUOI VEDERE LE DISPOSIZIONI?": "S/N"
1070 CALL KEY(0,KK,SS)
1080 IF SS=0 THEN 1070
1090 IF KK=83 THEN 1120
1100 IF KK=78 THEN 190
1110 GOTO 1070
1120 J=N-K+1
1130 GOSUB 2980
1140 C=0
1150 PRINT ":
1160 PRINT "PER RAPPRESENTARE GLI OGGETTI-"
1170 PRINT "I VUOI NUMERI O LETTERE?":
1180 PRINT "PREMI":
1190 PRINT "1. PER I NUMERI"
1200 PRINT "2. PER LE LETTERE"
1210 CALL KEY(0,KK,SS)
1220 IF SS=0 THEN 1210
1230 IF KK=49 THEN 1260
1240 IF KK=50 THEN 1430
1250 GOTO 1210
1260 PRINT ":
1270 CALL MCHAR(23,3,128,28)
1280 C=C+1
1290 IF C=D THEN 1330
1300 PRINT ":
1310 CALL MCHAR(23,3,128,28)
1320 PRINT ":
1330 GOSUB 2940
1340 GOTO 190
1350 PRINT TAB(5-LEN(STR(C))):C;" "
1360 FOR I=0 TO K-1
1370 PRINT P(N-I):
1380 NEXT I
1390 PRINT
1400 IF C/10<>INT(C/10) THEN 1430
1410 GOSUB 2940
1420 CALL MCHAR(23,3,32,28)
1430 GOSUB 2770
1440 GOTO 1280
1450 PRINT ":
1460 CALL MCHAR(23,3,128,28)
1470 C=C+1
1480 IF C=D THEN 1540
1490 PRINT ":
1500 CALL MCHAR(23,3,128,28)
1510 PRINT ":
1520 GOSUB 2940
1530 IF FLAG=1 THEN 1930 ELSE 190
1540 PRINT TAB(5-LEN(STR(C))):C;" "
1550 FOR I=0 TO K-1
1560 PRINT CHR$(P(N-I)+64):
1570 NEXT I
1580 PRINT
1590 IF C/10<>INT(C/10) THEN 1620
1600 GOSUB 2940
1610 CALL MCHAR(23,3,32,28)
1620 GOSUB 2770
1630 GOTO 1470
1640 CALL CLEAR
1650 PRINT "PREMI":
1660 PRINT "1. PER LE SPIEGAZIONI":
1670 PRINT "2. PER LE APPLICAZIONI":
1680 PRINT "3. PER GLI ANAGRAMMI.":
1690 CALL KEY(0,KK,SS)
1700 IF SS=0 THEN 1690
1710 IF KK=49 THEN 1730
1720 IF KK=50 THEN 2120
1730 IF KK=51 THEN 2330

```

```

1740 GOTO 1690
1750 CALL CLEAR
1760 PRINT "SIANO DATI TRE OGGETTI:":
1770 PRINT TAB(11);"A B C":
1780 PRINT "USANDOLI TUTTI E TRE, QUANTI"
1790 PRINT "GRUPPI SI POSSONO FORMARE IN"
1800 PRINT "MODI CHE DIFFERISCANO SOLO"
1810 PRINT "PER L'ORDINE DEGLI OGGETTI?":
1820 PRINT "VEDIAMO...":
1830 GOSUB 2940
1840 CALL MCHAR(23,3,32,28)
1850 N=3
1860 K=3
1870 J=1
1880 C=0
1890 D=6
1900 GOSUB 2980
1910 FLAG=1
1920 GOTO 1430
1930 CALL CLEAR
1940 FLAG=0
1950 PRINT "QUESTI GRUPPI SI CHIAMANO"
1960 PRINT "PERMUTAZIONI SEMPLICI, ESSI"
1970 PRINT "SONO LE DISPOSIZIONI DEI TRE"
1980 PRINT "OGGETTI PRESI A TRE A TRE":
1990 PRINT "P(N) = 3*2*1 = 3!":
2000 PRINT "IL NUMERO 3! SI DICE FATTO-"
2010 PRINT "ORIALE DEL NUMERO 3!":
2020 PRINT "GLI ANAGRAMMI SONO UN ESEM-"
2030 PRINT "PIO DI PERMUTAZIONI"
2040 PRINT ":
2050 PRINT "VUOI RIVEDERE LE PERMUTAZIO-"
2060 PRINT "NI? (S/N)"
2070 CALL KEY(0,KK,SS)
2080 IF SS=0 THEN 2070
2090 IF KK=83 THEN 1830
2100 IF KK=78 THEN 190
2110 GOTO 2070
2120 CALL CLEAR
2130 PRINT "QUANTI SONO GLI OGGETTI? (NON PIU' DI 10)":
2140 INPUT "N = ":N
2150 IF N<11 THEN 2180
2160 PRINT ":ATTENZIONE! DEVE ESSERE":N<=10, RIPROVA.":
2170 GOTO 2140
2180 PRINT ":
2190 PERM=N
2200 FOR I=1 TO N-1
2210 PERM=PERM*(N-I)
2220 NEXT I
2230 PRINT "IL NUMERO DELLE PERMUTAZIONI":E":
2240 PRINT "P=":STR$(N):"!":PERM:
2250 PRINT "VUOI VEDERE LE PERMUTAZIONI? (S/N)"
2260 CALL KEY(0,KK,SS)
2270 IF SS=0 THEN 2260
2280 IF KK=83 THEN 2310
2290 IF KK=78 THEN 190
2300 GOTO 2260
2310 K=N
2320 GOTO 1120
2330 CALL CLEAR
2340 PRINT "SCRIVI LA PAROLA DA ANAGRAM-"
2350 PRINT "MARE (NON PIU' DI 10 LETTERE)":
2360 INPUT PA$
2370 IF LEN(PA$)<=10 THEN 2430
2380 PRINT ":ATTENZIONE! IL PROGRAMMA"
2390 PRINT "NON E' DIMENSIONATO PER UNA"
2400 PRINT "PAROLA COSI' LUNGA, RIPROVA."
2410 PRINT ":
2420 GOTO 2360
2430 N=LEN(PA$)
2440 K=N
2450 J=1
2460 MAX=1
2470 FOR I=1 TO N
2480 AS(I)=SEG$(PA$,I,1)
2490 MAX=MAX*I
2500 NEXT I
2510 GOSUB 2980
2520 C=0
2530 PRINT ":ANAGRAMMI:":
2540 CALL MCHAR(23,3,128,28)
2550 PRINT
2560 C=C+1
2570 IF C=MAX THEN 2630
2580 PRINT
2590 CALL MCHAR(24,3,128,28)

```

```

2600 PRINT
2610 GOSUB 2940
2620 GOTO 190
2630 PRINT TAB(5-LEN(STR$(C)))C;"- ";
2640 FOR I=0 TO K-1
2650 PRINT A$(P(N-I))
2660 NEXT I
2670 PRINT
2680 IF C/10<>INT(C/10) THEN 2710
2690 GOSUB 2940
2700 CALL HCHAR(23,3,32,28)
2710 GOSUB 2770
2720 GOTO 2560
2730 REM
2740 REM          RICERCA
2750 REM          DISPOSIZIONI
2760 REM
2770 X=J
2780 CTR(X)=CTR(X)+1
2790 IF CTR(X)<=X THEN 2840
2800 CTR(X)=1
2810 X=X+1
2820 IF X>N THEN 2930
2830 GOTO 2780
2840 FOR W=X TO J STEP -1
2850 P(W)=P(W)+1
2860 IF W=N THEN 2920
2870 IF P(W)<=N THEN 2890
2880 P(W)=1
2890 FOR I=W+1 TO N
2900 IF P(I)=P(W) THEN 2850
2910 NEXT I
2920 NEXT W
2930 RETURN
2940 PRINT TAB(5)C;"- PREMI UN TASTO -"
2950 CALL KEY(0,KK,SS)
2960 IF SS=0 THEN 2950
2970 RETURN
2980 FOR I=J TO N
2990 CTR(I)=1
3000 P(I)=N-I-I
3010 NEXT I
3020 RETURN
3030 CALL CLEAR
3040 END

```

Il carattere 128 (linea 180) serve a creare una linea orizzontale che verrà posta all'inizio e al termine della stampa delle disposizioni. Dalla riga 400 alla 680 si trovano le spiegazioni sul significato delle disposizioni semplici e, a titolo d'esempio, vengono stampate le 24 disposizioni semplici relative a quattro oggetti presi a tre a tre. La ricerca dei gruppi è svolta dalle routine 2980 e 2770, le stesse utilizzate quando sarà l'utente a stabilire il numero degli oggetti e la classe K; semplicemente, prima di richiamare le routine, viene posto $N=4$ e $K=3$ (linee 520-530). Le istruzioni 650-670 servono a stampare due gruppi per ogni riga in modo che al termine siano visibili tutte e 24 le disposizioni per meglio esaminarle. Alla linea 690 inizia la parte applicativa. Il ciclo FOR-NEXT (1010-1030) calcola il numero delle disposizioni che viene stampato con le istruzioni relative. I gruppi che costituiscono le disposizioni sono scritti solo su richiesta e si utilizzano due differenti cicli di stampa a seconda che gli oggetti siano rappresentati con cifre o lettere; nel primo caso l'istruzione PRINT è quella di riga 1370, nel secondo quella di riga 1560. Le condizioni poste alle linee 1400 e 1590 servono a scrivere i dati in blocchi di dieci; scritto un blocco il programma si ferma e riparte alla pressione di un tasto qualsiasi. Lo scopo è quello di dare all'utente il tempo di esaminare quanto appare sullo schermo, senza costringerlo a bloccare l'ese-

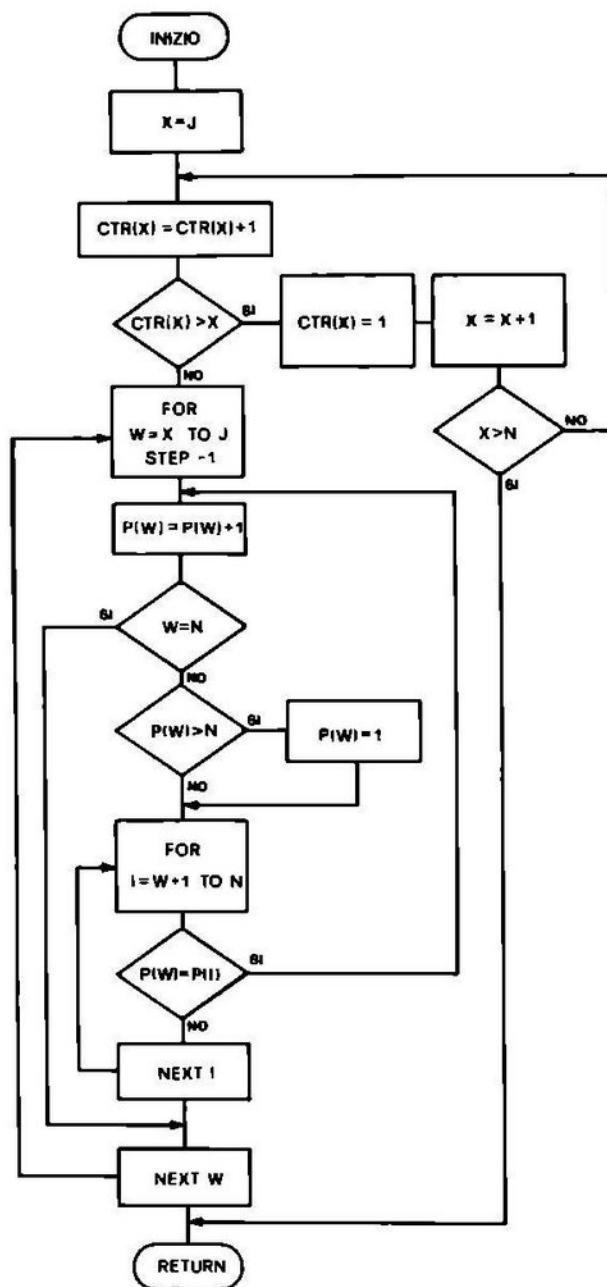


Fig. 16

cuzione premendo FCTN(4) per poi proseguire con il comando immediato con < ENTER >. La variabile FLAG di linea 1530 è necessaria poiché quel ciclo di stampa è utilizzato anche dalla sezione 2 del programma per stampare un esempio di permutazioni. PERM è il numero delle permutazioni ed è il fattoriale di N (2190-2220). Poiché queste sono un caso particolare di disposizioni in cui la classe K è uguale al numero degli og-

RUN

PREMI:

1. PER LE DISPOSIZIONI SEMPLICI DI N OGGETTI
2. PER LE PERMUTAZIONI DI N OGGETTI
3. PER FINIRE

<1>

PREMI:

1. PER LE SPIEGAZIONI
2. PER LE APPLICAZIONI

<2>

QUANTI SONO GLI OGGETTI?
(NON PIU' DI 10)

N = 5

<ENTER>

QUANTI OGGETTI FORMANO OGNI GRUPPO?

K = 3

<ENTER>

IL NUMERO DELLE DISPOSIZIONI E':

$D(5,3) = 60$

VUOI VEDERE LE DISPOSIZIONI?
(S/N)

<S>

PER RAPPRESENTARE GLI OGGETTI VUOI NUMERI O LETTERE?

PREMI:

1. PER I NUMERI
2. PER LE LETTERE

<2>

1 - A B C
2 - A B D
3 - A B E
4 - A C B
5 - A C D

58 - E D C
59 - E D A
60 - E D B

- PREMI UN TASTO -

getti N, basta porre questa condizione (linea 2310) e rimandare il programma alla sezione precedente (linea 2320).

L'ultima parte è dedicata alla scrittura degli anagrammi di parole formate da non più di 10 lettere. L'anagramma consiste nel permutare le lettere di una parola. Da un punto di vista enigmistico l'anagramma dovrebbe consistere in un secondo termine di senso compiuto, così come da "caro" si può ricavare "arco", oppure "orca"; ma il nostro TI 99/4A, che non possiede il fine senso letterario del computer, stamperà indiscriminatamente tutte le permutazioni comprese "roca" e "roac".

Il vero cuore del programma è la routine 2770-2930 (vedi il diagramma di figura 16), essa va utilizzata dopo che le variabili con indice CTR e P sono state inizializzate nell'altra routine di linea 2980.

CTR conta quante volte è stato cambiato l'elemento che occupa una certa posizione nel gruppo. L'istruzione 2790 controlla che un certo elemento non venga cambiato più del necessario; se un elemento ha due gradi di libertà, potrà cambiare solo due volte, dopo di che ne verrà cambiato un altro; per questa ragione CTR(2) non può mai essere maggiore di due, analogamente CTR(3) non può essere mai maggiore di tre, ecc. L'oggetto che fa parte di un gruppo viene cambiato con l'istruzione 2850, il ciclo FOR-NEXT (2890-2910) controlla che non si siano verificate ripetizioni.

La memoria utilizzata è di 6,1 Kbyte.

Vediamo ora un esempio pratico di utilizzazione. Per la stampa delle 60 disposizioni il TI 99/4A ha impiegato meno di 1 minuto.

Combinazioni semplici

Il programma è strutturato come il precedente: la prima parte è riservata alle spiegazioni, la seconda alle applicazioni. Le combinazioni semplici costituiscono un sottoinsieme delle disposizioni semplici essendo formate da gruppi di n oggetti presi a k a k in modo tale che differiscono solo per almeno un oggetto ma non per l'ordine in cui gli oggetti sono disposti. Il numero delle combinazioni è dato da:

$$C_{n,k} = \frac{D_{n,k}}{k!}$$

Nel caso particolare in cui sia $n=5$ e $k=3$, il numero delle combinazioni semplici è:

$$C_{5,3} = \frac{5 \times 4 \times 3}{3 \times 2 \times 1} = 10$$

Un'originale routine, di cui si riporta il diagramma a blocchi, fornisce, oltre al numero delle combinazioni, anche la loro stampa per ogni valore di n inferiore o uguale a venti. Per $n > 20$ è necessario dimensionare diversamente il programma nell'istruzione DIM ed in quella che controlla se $n \leq 20$.

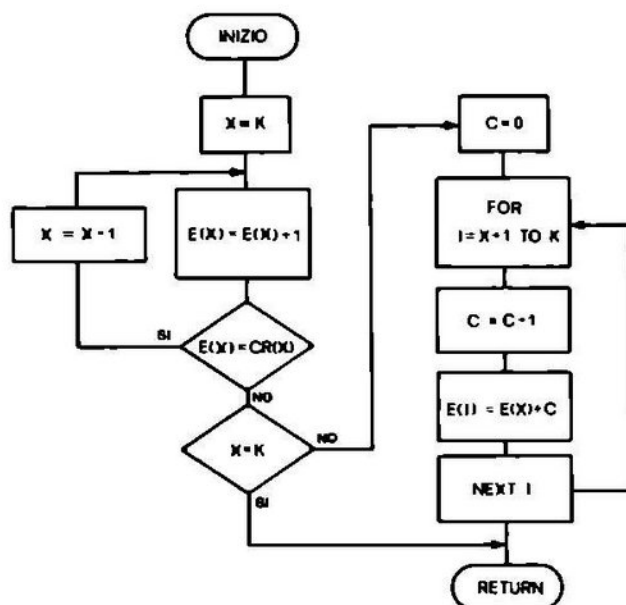


Fig. 17

Il concetto base consiste nel caricare i valori numerici da attribuire ad ogni elemento di un singolo gruppo in una variabile con indice $E(X)$ compreso tra 1 e k . Il vettore $E(X)$ viene inizializzato con lo stesso valore dell'indice: $E(1)=1$, $E(2)=2$, $E(3)=3$, ecc.; si aumenta la variabile con indice maggiore finché il suo valore è uguale ad n , dopo di che si incrementa il valore della variabile con indice immediatamente inferiore. Nell'esempio, già citato, di combinazioni di 5 oggetti della classe 3, la routine attribuisce successivamente al vettore $E(X)$ i seguenti valori:

	E(1)	E(2)	E(3)		E(1)	E(2)	E(3)
1 —	1	2	3	6 —	1	4	5
2 —	1	2	4	7 —	2	3	4
3 —	1	2	5	8 —	2	3	5
4 —	1	3	4	9 —	2	4	5
5 —	1	3	5	10 —	3	4	5

Come si vede, l'elemento E(1) può raggiungere al massimo il valore 3, E(2) al massimo il valore 4, ecc.; tali massimi sono memorizzati nel vettore CR(X). Ogniqualvolta un elemento raggiunge il suo valore massimo si passa ad incrementare l'elemento con indice immediatamente inferiore mentre a tutti i seguenti si attribuiscono valori progressivi.

In fase di stampa, per evitare uno scrolling incontrollato, dopo ogni dieci combinazioni il programma si arresta in attesa che venga premuto un tasto.

```

100 REM COMBINAZIONI SEMPLICI
110 REM *****
120 REM di: Sergio Borsani
130 REM versione: TI BASIC
140 CALL CLEAR
150 DIM E(20), CR(20)
160 PRINT TAB(13); "COMBINAZIONI SEMPLICI":;
170 CALL CHAR(128, "00000000FF")
180 FOR T=1 TO 500
190 NEXT T
200 CALL CLEAR
210 PRINT "PREMI:":
220 PRINT "1. PER LE SPIEGAZIONI":
230 PRINT "2. PER LE APPLICAZIONI":
240 PRINT "3. PER FINIRE":
250 CALL KEY(0, KEY, STATO)
260 IF STATO=0 THEN 250
270 IF KEY=49 THEN 310
280 IF KEY=50 THEN 940
290 IF KEY=51 THEN 2070
300 GOTO 250
310 CALL CLEAR
320 PRINT "SIANO DATI CINQUE OGGETTI":
330 PRINT TAB(10); "A B C D E":
340 PRINT "CON ESSI SI VOGLIONO FORMARE"
350 PRINT "GRUPPI DI TRE ELEMENTI CHE"
360 PRINT "DIFFERISCANO ALMENO PER UN"
370 PRINT "OGGETTO":
380 PRINT "SCRIVIAMOLI...":
390 GOSUB 2020
400 CALL MCHAR(24, 3, 128, 28)
410 PRINT
420 PRINT "1 - ABC": "2 - ABD": "3 - ABE"
430 PRINT "4 - ACD": "5 - ACE": "6 - ADE"
440 PRINT "7 - BCD": "8 - BCE": "9 - BDE"
450 PRINT "10 - CDE":
460 CALL MCHAR(23, 3, 128, 28)
470 PRINT :
480 PRINT "QUESTI GRUPPI COSTITUISCONO"
490 PRINT "LE COMBINAZIONI SEMPLICI DI"
500 PRINT "5 OGGETTI PRESI A 3 A 3, 0"
510 PRINT "DELLA CLASSE 3.":
520 GOSUB 2020
530 CALL CLEAR
540 PRINT "PERMUTANDO OGNI GRUPPO SI"
550 PRINT "OTTIENE LA TOTALITA' DELLE"
560 PRINT "DISPOSIZIONI SEMPLICI DI 5"
570 PRINT "OGGETTI DELLA CLASSE 3.":
580 PRINT "LE COMBINAZIONI SONO PERTAN-"
590 PRINT "TO UN SOTTOINSIEME DELLE DI-"
600 PRINT "SPOSIZIONI.":
610 PRINT "NEL NOSTRO ESEMPIO RISULTA:":
620 PRINT "D(5,3)=5*4*3"
630 PRINT "P(3)=3!=3*2*1":
640 PRINT TAB(12); "5*4*3"
650 PRINT TAB(15); "C(5,3)=:"
660 FOR I=1 TO 3
670 PRINT CHR(128);
680 NEXT I
690 PRINT "10"
700 PRINT TAB(12); "3*2*1":
710 GOSUB 2020
720 CALL CLEAR
730 PRINT "PIU' IN GENERALE, IL NUMERO"
740 PRINT "DELLE COMBINAZIONI SEMPLICI"
750 PRINT "DI N OGGETTI DELLA CLASSE K"
760 PRINT "E'":
770 PRINT TAB(15); "D(N,K)"
780 PRINT TAB(8); "C(N,K)=:"

```

```

790 FOR I=1 TO 6
800 PRINT CHR$(128);
810 NEXT I
820 PRINT
830 PRINT TAB(17);"K!":=1
840 PRINT "DIVERO":=1
850 PRINT TAB(8);"N(N-1)(N-2)...(N-K+1)"
860 PRINT "C(N,K)="+1
870 FOR I=1 TO 21
880 PRINT CHR$(128);
890 NEXT I
900 PRINT
910 PRINT TAB(17);"K!":=1
920 GOSUB 2020
930 GOTO 200
940 CALL CLEAR
950 PRINT "SCRIVI IL NUMERO DI OGGETTI":=1
960 INPUT "N = ";N
970 N=ABS(INT(N))
980 PRINT "DI QUALE CLASSE SONO LE COM-":="BINAZIONI? (K<=N)":=1
1000 INPUT "K = ";K
1010 IF K<=N THEN 1040
1020 PRINT "ERRORE! RIPROVA":=1
1030 GOTO 1000
1040 GOSUB 1410
1050 GOSUB 1490
1060 PRINT "IL NUMERO DELLE COMBINAZIONI"
1080 PRINT "SEMPLICI E":=1
1090 PRINT TAB(8);"C =";CNK
1100 PRINT "SE STATO=0 THEN 1140"
1110 IF N<21 THEN 1140
1120 GOSUB 2020
1130 GOTO 200
1140 PRINT "VUOI VEDERE LE COMBINAZIONI?":="(S/N)"
1150 CALL KEY(0,KEY,STATO)
1160 IF STATO=0 THEN 1150
1170 IF KEY=83 THEN 1200
1180 IF KEY=78 THEN 200
1190 GOTO 1150
1200 PRINT "VUOI CHE GLI OGGETTI VENGANO"
1220 PRINT "RAPPRESENTATI CON NUMERI O"
1230 PRINT "LETTERE?":=1
1240 PRINT "PREMI":=1
1250 PRINT "1. PER I NUMERI":=1
1260 PRINT "2. PER LE LETTERE":=1
1270 CALL KEY(0,KEY,STATO)
1280 IF STATO=0 THEN 1270
1290 IF KEY=49 THEN 1320
1300 IF KEY=50 THEN 1340
1310 GOTO 1270
1320 FLAG=1
1330 GOTO 1350
1340 FLAG=2
1350 GOSUB 1750
1360 GOSUB 2020
1370 GOTO 200
1380 REM
1390 REM INIZIALIZZAZIONE
1400 REM
1410 FOR I=1 TO K
1420 CR(I)=N-K+I
1430 E(I)=I
1440 NEXT I
1450 RETURN
1460 REM
1470 REM NUM. COMBINAZIONI
1480 REM
1490 CNK=N
1500 KK=1
1510 FOR I=1 TO K-1
1520 CNK=CNK*(N-I)
1530 KK=KK*(I+1)
1540 NEXT I
1550 CNK=CNK/KK
1560 RETURN
1570 REM
1580 REM COMBINAZIONI
1590 REM
1600 X=K
1610 E(X)=E(X)+1
1620 IF E(X)<=CR(X) THEN 1650
1630 X=X-1
1640 GOTO 1610

```

```

1650 IF X=K THEN 1710
1660 C=0
1670 FOR I=X+1 TO K
1680 C=C+1
1690 E(I)=E(X)+C
1700 NEXT I
1710 RETURN
1720 REM
1730 REM      STAMPA
1740 REM
1750 CTR=0
1760 PRINT "::::"
1770 PRINT "COMBINAZIONI SEMPLICI."
1780 CALL HCHAR(24,3,128,28)
1790 PRINT
1800 CTR=CTR+1
1810 IF CTR>CNK THEN 1950
1820 PRINT TAB(6-LEN(STR$(CTR)));CTR;" - ";
1830 FOR I=1 TO K
1840 IF FLAG=1 THEN 1870
1850 PRINT CHR$(64+E(I));
1860 GOTO 1880
1870 PRINT STR$(E(I));
1880 NEXT I
1890 PRINT
1900 IF CTR/10<>INT(CTR/10) THEN 1920
1910 GOSUB 1990
1920 IF CTR=CNK THEN 1950
1930 GOSUB 1570
1940 GOTO 1800
1950 CALL HCHAR(24,3,128,28)
1960 PRINT
1970 PRINT TAB(25);"FINE":;
1980 RETURN
1990 REM
2000 REM      PREMI UN TASTO
2010 REM
2020 PRINT "      - PREMI UN TASTO -"
2030 CALL KEY(0,KEY,STAT)
2040 IF STAT=0 THEN 2030
2050 CALL HCHAR(23,3,32,28)
2060 RETURN
2070 CALL CLEAR
2080 END

```

Seguono alcune brevi note al listato.

150: il dimensionamento riguarda gli oggetti con i quali si vogliono ottenere le combinazioni (variabile con indice E) e una variabile di controllo (CR) per la formazione dei gruppi.

170: il carattere 128 traccia una linea orizzontale usata anche come linea di frazione.

200-300: menù che rinvia alle due sezioni del programma; le spiegazioni occupano le righe 310-930, esse hanno solo uno scopo espositivo e non necessitano di commenti; le applicazioni vanno da riga 940 a riga 1370; le istruzioni seguenti costituiscono delle routine.

960: il numero degli oggetti viene memorizzato nella variabile N; la linea seguente protegge il programma nell'eventualità che venga digitato un numero negativo con cifre decimali.

1000: K è la classe delle combinazioni, ovvero il numero degli oggetti che formano ogni gruppo.

1110: le combinazioni non vengono stampate se N è maggiore di 20.

1320-1340: la variabile FLAG condiziona la stampa di numeri (FLAG=1) o lettere (FLAG=2) per rappresentare gli oggetti.

1410-1450: sottoprogramma inizializzazione. Le variabili di controllo CR(I) assumono i valori iniziali N-K+1, vale a dire che, se N=5 e K=3,

le tre variabili di controllo avranno i valori 3, 4 e 5. Il vettore E(i) contiene invece gli oggetti rappresentati inizialmente dai numeri progressivi 1, 2, 3,... fino a K.

1490-1560: sottoprogramma per il calcolo del numero delle combinazioni. L'istruzione 1520 calcola il numero delle disposizioni, mentre la 1530 calcola il fattoriale di K.

Alla linea 1560 si calcola il numero delle combinazioni; per risparmiare memoria usiamo sempre la variabile CNK poiché non è necessario conservare il numero delle disposizioni.

1600-1710: routine per creare le combinazioni (vedi diagramma a blocchi, figura 17).

1750-1980: sottoprogramma per la stampa delle combinazioni. L'istruzione 1820 ha lo scopo di allineare a destra i numeri progressivi. In base al valore di FLAG (linea 1840) si stampa una lettera (linea 1850) o un numero (linea 1870). L'istruzione 1900 ferma il programma ogni 10 combinazioni; esso riprende alla pressione di un tasto.

La memoria occupata è di 4,3 Kbyte.

Vediamo ora un'applicazione riguardante il gioco del lotto. Proponiamoci di voler calcolare la probabilità di indovinare un terno in una ruota (le ruote sono le dieci città: Bari, Cagliari, Firenze, Genova, Milano, Napoli, Palermo, Roma, Torino e Venezia). Il metodo potrà essere esteso agli ambi, alle quaterne ecc.

I terni sono combinazioni semplici di 90 oggetti (numeri) presi a tre a tre, pertanto il numero di eventi possibili è:

$$C\left(\begin{matrix} 90 \\ 3 \end{matrix}\right) = 117480$$

RUN

```

SCRIVI IL NUMERO DI OGGETTI
N = 90                                <ENTER>
DI QUALE CLASSE SONO LE
COMBINAZIONI? (K<=N)
K = 3                                <ENTER>
IL NUMERO DELLE COMBINAZIONI
SEMPLICI E':
      C = 117480
- PREMI UN TASTO -
  
```

Ma attenzione! Nel calcolare la probabilità si deve tener presente che i numeri estratti sono 5; essi, presi a tre a tre, danno luogo a 10 combinazioni.

Dunque le probabilità di fare terno sono:

$$P = C\left(\begin{matrix} 5 \\ 3 \end{matrix}\right) \times \frac{1}{C\left(\begin{matrix} 90 \\ 3 \end{matrix}\right)} = 10 \times \frac{1}{117480} = \frac{1}{11748}$$

Lo stato paga solo 4250 volte la posta!

Distribuzione normale o gaussiana

Alcuni concetti statistici sono applicati anche nell'ambito della misura di una grandezza fisica. È questo un campo nel quale molti presuppongono che si possano raggiungere delle conoscenze con la massima accuratezza e precisione, ogni misura è soggetta a limitazioni e ad errori casuali. Se, ad esempio, volessimo misurare il periodo di oscillazione di un pendolo, ripetendo l'esperienza più volte, otterremo spesso valori diversi e saremo infine portati a ritenere come più probabile il valore medio.

$$\bar{x} = \frac{1}{n} (x_1 + x_2 + x_3 + \dots + x_n)$$

Per avere indicazioni sulla distribuzione dei dati attorno alla media, è comunemente usato lo scarto quadratico medio.

$$\mu = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$$

Valori piccoli di μ significano che i risultati delle misure si addensano attorno al valor medio, contrariamente i risultati sono dispersi in un intervallo più ampio.

Quando il numero delle prove è elevato (alcune centinaia), riportando in un istogramma le frequenze relative ai vari risultati ottenuti, si ottiene un grafico simile a quello di figura 18.

Il programma che segue consente di dimostrare che una tale distribuzione è in buon accordo con una distribuzione normale o gaussiana.

Questa, in termini matematici, è data dalla funzione:

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-1/2(x-\mu)^2/\sigma^2}$$

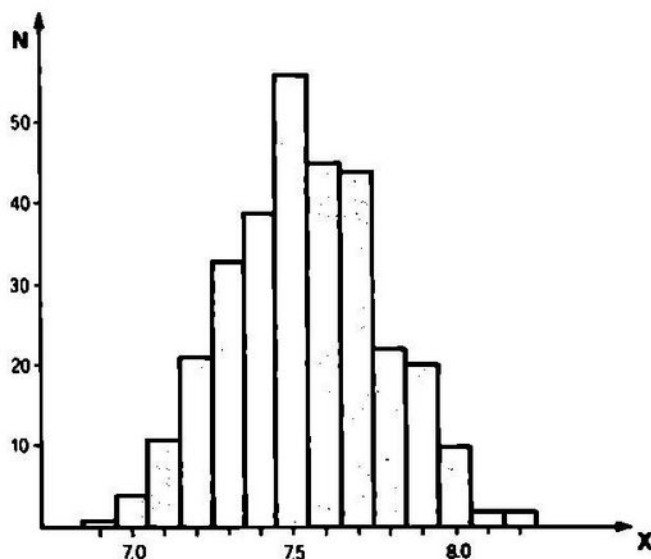


Fig. 18

dove la quantità σ è chiamata deviazione standard e, per un numero elevato di misure, coincide con lo scarto quadratico medio.

La funzione $f(x)$ indica quale sia la probabilità che una misura dia per risultato un valore x . Il grafico della funzione è una curva a campana. L'area della superficie racchiusa dalla curva e dall'asse delle x è uguale ad 1, mentre l'area delimitata sotto la curva tra due valori a e b è < 1 e rappresenta la probabilità che x sia compreso tra a e b ; ricordando il significato geometrico dell'integrale, tale probabilità è:

$$P_a = \int_a^b f(x) dx$$

Ad esempio si dimostra che una misura ha una probabilità del 68,3% di cadere nell'intervallo $\bar{x} - \sigma$, $\bar{x} + \sigma$. Vedi figura 19.

Ora vediamo come viene condotto il confronto tra la distribuzione di dati sperimentali ed una distribuzione gaussiana.

Innanzitutto, è ovvio, bisogna disporre di dati sperimentali. Questi non possono essere sostituiti con valori casuali generati dal computer, in quanto essendo questi ultimi equiprobabili non godrebbero di una distribuzione normale di probabilità.

Tornando all'esempio della misura del periodo di oscillazione di un pendolo, è necessario ripetere l'esperienza numerosissime volte (200 o 300) e segnare quante volte si ripetono gli stessi risultati. Per avviare il pro-

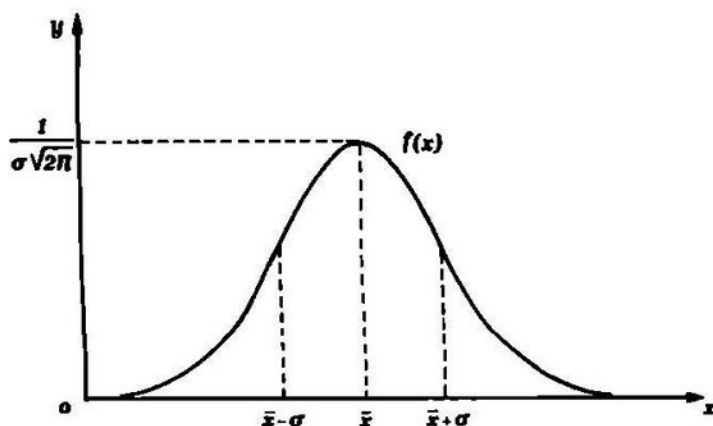


Fig. 19

gramma è sufficiente scrivere il valore minimo, quanti diversi risultati si sono ottenuti (fino ad un massimo di 30) e qual è la minima differenza tra un risultato e l'altro (ad es.: 1/100 di secondo). Il computer chiederà quante volte sono stati ottenuti i singoli risultati. Dopo questa operazione si può accedere a diverse sezioni del programma, tutte indipendenti una dall'altra e che pertanto possono essere eseguite in modo non sequenziale.

Le funzioni svolte sono le seguenti: (1) tabulazione dei risultati e loro rappresentazione in forma di istogramma; (2) calcolo del valore medio; (3) tabulazione degli scarti dalla media; (4) tabulazione degli scarti quadratici; (5) calcolo dello scarto quadratico medio; (6) calcolo delle probabilità che un risultato cada in un dato intervallo; (7) tabulazione delle frequenze teoriche previste per una distribuzione gaussiana; (8) confronto tra l'istogramma dei risultati sperimentali e quello dei valori previsti in base alla funzione di distribuzione normale.

Le principali variabili utilizzate nel programma sono: MIN, minimo valore ottenuto nella serie di misure; DX, scarto minimo tra due risultati; N, numero di diversi risultati ottenuti.

I singoli risultati sono pertanto: MIN, MIN + DX, MIN + 2DX, MIN + 3DX, ecc.; il numero delle volte in cui si sono ottenuti questi valori viene caricato nella variabile con indice A(i).

La base è 0, così A(0) corrisponde al risultato MIN, A(1) a MIN + DX, A(2) al successivo, e così via.

Il totale delle misure risulta:

$$M = \sum_{i=0}^{N-1} A(i)$$

Per il valore medio si usa la relazione:

$$\bar{x} = \frac{1}{M} \sum_{i=0}^{N-1} (MIN + i \times DX) \times A(i)$$

Naturalmente questa sommatoria si ottiene con le istruzioni Basic.

```
MEDIA=0
FOR I=0 TO N-1
MEDIA=MEDIA+(MIN+I*DX)*A(I)
NEXT I
```

Lo scarto quadratico medio si trova con una formula analoga a quella data in precedenza

$$\mu = \sqrt{\frac{1}{M} \sum_{i=0}^{N-1} A(i) \times (MIN + i \times DX - MEDIA)^2}$$

Chiamato per semplicità Z_i ogni risultato precedentemente espresso nella forma $MIN + i \times DX$, la probabilità di ottenere l' i -esimo valore è calcolata con la formula:

$$P_i = \frac{DX}{\mu \sqrt{2\pi}} e^{-1/2(Z_i - \bar{x})^2/\mu^2}$$

Le frequenze teoriche, memorizzate nel vettore $C(i)$, sono allora: $M \times P_i$. Il programma fornisce la tabulazione dei vettori $A(i)$ e $C(i)$, cioè dei valori sperimentali e di quelli previsti teoricamente, e il confronto tra gli istogrammi. Il computer dimensiona automaticamente i grafici e li mostra contemporaneamente sullo schermo in due diversi colori. Se l'esperienza è ben condotta ed il numero delle misure è sufficientemente grande, si deve notare un accordo tra i due istogrammi.

Infine, il calcolo delle probabilità che un valore cada in un dato intervallo è ottenuto integrando la funzione $f(x)$ con il metodo di Simpson. L'intervallo va specificato nella forma $\pm h$. Attribuendo ad x il valore medio e ad h lo scarto quadratico medio, si ha l'immediata conferma che la probabilità è il 68,3%.

Per permettere di utilizzare il programma anche a chi non dispone di adeguati risultati sperimentali, si riporta la seguente tabella con i valori ottenuti nel corso di una esercitazione di fisica, tenuta presso l'Università di Padova. I risultati, riguardanti più di trecento misure, si riferiscono alla lunghezza di una colonna di sferette contenute in un tubo di plexiglass e sono dati a meno di una costante.

Z_i (mm)	$A(i)$
6.9	1
7.0	4
7.1	11
7.2	21
7.3	33
7.4	39
7.5	56
7.6	45
7.7	44
7.8	22
7.9	20
8.0	10
8.1	2
8.2	2

Ai fini di ciò che viene richiesto dal programma è utile ricordare che il valore minimo è 6.9, che lo scarto tra un valore e il successivo è 0.1 mm, che i risultati ottenuti sono 14 e che $A(i)$ sono le frequenze relative ai singoli risultati.

```

100 REM  DISTRIBUZIONE GAUSSIANA
110 REM  *****
120 REM  DI: SERGIO BORSANI
130 REM  VERSIONE: TI BASIC
140 CALL CLEAR
150 PRINT TAB(8);"VERIFICA DELLA";TAB(4);"DISTRIBUZIONE GAUSSIANA":::::::::
160 DIM A(30),B(30),C(30)
170 DATA 000000000000007F
180 DATA 00000000000007F7F
190 DATA 000000000007F7F7F
200 DATA 0000000007F7F7F7F
210 DATA 0000007F7F7F7F7F7F
220 DATA 00007F7F7F7F7F7F7F
230 DATA 007F7F7F7F7F7F7F7F
240 DATA 7F7F7F7F7F7F7F7F7F
250 FOR I=1 TO 8
260 READ CAR$
270 CALL CHAR(127+I,CAR$)
280 CALL CHAR(135+I,CAR$)
290 NEXT I
300 CALL CHAR(144,"FF")
310 CALL CHAR(145,"00000000FF")
320 CALL COLOR(13,2,1)
330 CALL COLOR(14,2,1)
340 CALL CHAR(146,"002424243A202040")
350 CALL CLEAR
360 PRINT " LA MISURA DI UNA GRANDEZZA": "FISICA E' SOGGETTA AD ERRORI"
370 PRINT "CASUALI.": " RIPETENDO MOLTE VOLTE LA"
380 PRINT "MISURA, I RISULTATI SODDI-": "SFANO AD UNA DISTRIBUZIONE"
390 PRINT "NORMALE O GAUSSIANA.":::::::::
400 GOSUB 3550
410 CALL CLEAR
420 PRINT " PER UNA VERIFICA, SCRIVERE": "LA PIU' PICCOLA MISURA OTTE-"
430 PRINT "NUTA, L'AMPIEZZA DEGLI IN-": "TERVALLI IN CUI CADONO LE"
440 PRINT "MISURE E IL NUMERO DEGLI IN-": "TERVALLI.":::
450 INPUT "VALORE MINIMO = ":MIN
460 PRINT ": "AMPIEZZA"
470 INPUT "DEGLI INTERVALLI = ":DX
480 PRINT ": "NUMERO"

```

```

490 INPUT "DEGLI INTERVALLI = ":N
500 IF N<=30 THEN 530
510 PRINT "SPIACENTE, NON PIU' DI 30.":
520 GOTO 480
530 PRINT "!!!"
540 GOSUB 3550
550 CALL CLEAR
560 PRINT "ORA SCRIVI LE FREQUENZE RELATIVE AD OGNI INTERVALLO":
570 PRINT "NUM. - VALORE FREQUENZA"
580 CALL HCHAR(24,3,145,28)
590 PRINT
600 M=0
610 MAX=0
620 FOR I=0 TO N-1
630 PRINT STR$(I+1); " " ; TAB(8); MIN+DX*I; TAB(17);
640 INPUT A(I)
650 M=M+A(I)
660 IF A(I)<=MAX THEN 680
670 MAX=A(I)
680 NEXT I
690 CALL CLEAR
700 PRINT "PER LA SCELTA PREMI IL": "NUMERO CORRISPONDENTE":
710 PRINT "1. FREQUENZE SPERIMENTALI"
720 PRINT "2. VALORE MEDIO"
730 PRINT "3. SCARTI DALLA MEDIA"
740 PRINT "4. SCARTI QUADRATICI"
750 PRINT "5. SCARTO QUADRATICO MEDIO"
760 PRINT "6. PROBABILITA' DI UN EVENTO": "IN UN DATO INTERVALLO"
770 PRINT "7. FREQUENZE TEORICHE PER": "UNA DISTRIBUZIONE NORMALE"
780 PRINT "8. CONFRONTO TRA LE FREQUENZE SPERIMENTALI E LA DI-"
790 PRINT "STRIBUZIONE GAUSSIANA": "9. FINE PROGRAMMA":
800 CALL KEY(0,K,S)
810 IF B=0 THEN 800
820 IF (K<49)+(K>58) THEN 800
830 K=K-48
840 ON K GOTO 850,1200,1340,1600,1810,1910,2240,2580,3590
850 CALL CLEAR
860 PRINT "FREQUENZE SPERIMENTALI"
870 CALL HCHAR(24,3,145,28)
880 PRINT
890 PRINT "NUM. - VALORE - FREQUENZA"
900 CALL HCHAR(24,3,145,28)
910 PRINT
920 FOR I=0 TO N-1
930 PRINT STR$(I+1); " " ; TAB(8); MIN+I*DX; TAB(22-LEN(STR$(A(I))))A(I)
940 IF (I+1)/10<>INT((I+1)/10) THEN 970
950 GOSUB 3550
960 CALL HCHAR(23,3,32,28)
970 NEXT I
980 CALL HCHAR(24,3,145,28)
990 PRINT
1000 PRINT "NUMERO TOTALE": "DELLE MISURE":
1010 PRINT TAB(22-LEN(STR$(M)))M
1020 CALL HCHAR(24,3,145,28)
1030 PRINT "VUOI VEDERE L'ISTOGRAMMA?": "(B/N)"
1040 CALL KEY(0,K,S)
1050 IF B=0 THEN 1040
1060 IF K=78 THEN 690
1070 IF K=83 THEN 1090
1080 GOTO 900
1090 INIT=(30-N)/2
1100 FOR I=0 TO N-1
1110 B(I)=A(I)
1120 NEXT I
1130 U=MAX/20
1140 ULT=N-1
1150 COL=135
1160 CALL CLEAR
1170 GOSUB 3260
1180 GOSUB 3550
1190 GOTO 690
1200 CALL CLEAR
1210 PRINT "VALORE MEDIO"
1220 CALL HCHAR(24,3,145,28)
1230 PRINT "!!!"
1240 GOSUB 3370
1250 PRINT "NUMERO TOTALE DELLE MISURE":
1260 PRINT "N =":M:
1270 PRINT "VALORE MEDIO":
1280 PRINT "X =":INT(1000*MEDIA)/1000:
1290 CALL HCHAR(24,3,145,28)
1310 PRINT
1320 GOSUB 3550
1330 GOTO 690
1340 CALL CLEAR
1350 PRINT "SCARTI DALLA MEDIA"

```

```

1360 CALL MCHAR(24,3,145,28)
1370 PRINT
1380 IF FF1=1 THEN 1400
1390 GOSUB 3370
1400 PRINT "NUM. - VALORE - SCARTO"
1410 CALL MCHAR(24,3,145,28)
1420 PRINT
1430 SOMMA=0
1440 FOR I=0 TO N-1
1450 V=MIN+I*DX
1460 SCAR=ABS(V-MEDIA)
1470 PRINT STR$(I+1);";";TAB(8);V;TAB(17);INT(100*(SCAR+.005))/100
1480 SOMMA=SOMMA+SCAR*A(I)
1490 IF (I+1)/10<>INT((I+1)/10) THEN 1520
1500 GOSUB 3350
1510 CALL MCHAR(23,3,32,28)
1520 NEXT I
1530 CALL MCHAR(24,3,145,28)
1540 PRINT
1550 PRINT "SCARTO MEDIO";TAB(17);INT(100*SOMMA/M)/100
1560 CALL MCHAR(24,3,145,28)
1570 PRINT ";;;"
1580 GOSUB 3350
1590 GOTO 690
1600 CALL CLEAR
1610 PRINT "SCARTI QUADRATICI"
1620 CALL MCHAR(24,3,145,28)
1630 PRINT
1640 PRINT "NUM. - VALORE - SCARTO";TAB(17);"QUADRATICO"
1650 CALL MCHAR(24,3,145,28)
1660 PRINT
1670 IF FF1=1 THEN 1690
1680 GOSUB 3370
1690 FOR I=0 TO N-1
1700 V=MIN+I*DX
1710 SCAR2=(V-MEDIA)^2
1720 PRINT STR$(I+1);";";TAB(8);V;TAB(17);INT(1000*SCAR2)/1000
1730 IF (I+1)/10<>INT((I+1)/10) THEN 1760
1740 GOSUB 3350
1750 CALL MCHAR(23,3,32,28)
1760 NEXT I
1770 CALL MCHAR(24,3,145,28)
1780 PRINT ";;;"
1790 GOSUB 3350
1800 GOTO 690
1810 CALL CLEAR
1820 PRINT "SCARTO QUADRATICO MEDIO"
1830 CALL MCHAR(24,3,145,28)
1840 PRINT ";;;"
1850 GOSUB 3440
1860 PRINT TAB(6);CHR$(146);" =";INT(1000*SQM)/1000;";"
1870 CALL MCHAR(24,3,145,28)
1880 PRINT ";;;"
1890 GOSUB 3350
1900 GOTO 690
1910 CALL CLEAR
1920 PRINT "PROBABILITA' "IN UN DATO INTERVALLO"
1930 CALL MCHAR(24,3,145,28)
1940 PRINT ";;;"
1950 PRINT "L'INTERVALLO E' DATO NELLA:"FORMA (X-H),(X+H).":
1960 INPUT "X = ";XI
1970 INPUT "H = ";HI
1980 PRINT ";;;"
1990 IF FF2=1 THEN 2010
2000 GOSUB 3440
2010 DEF P(X)=.39894/SQM*EXP((-1/2)*((X-MEDIA)/SQM)^2)
2020 NT=0
2030 H=HI/10
2040 ST=0
2050 FOR I=1 TO 19 STEP 2
2060 X=(XI-HI)+(I*H)
2070 ST=ST+P(X)
2080 NEXT I
2090 NT=ST*4
2100 ST=0
2110 FOR I=2 TO 18 STEP 2
2120 X=(XI-HI)+(I*H)
2130 ST=ST+P(X)
2140 NEXT I
2150 NT=NT+2*ST+P(XI-HI)+P(XI+HI)
2160 NT=NT*H/3
2170 PRINT "PROBABILITA' CHE SI VERIFI-": "CHI UN EVENTO NELL'INTERVALLO"
2180 PRINT "LO CONSIDERATO: ";
2190 PRINT TAB(8);"P =";INT(1000*(NT+.005))/10;"%":
2200 CALL MCHAR(24,3,145,28)
2210 PRINT ";;;"

```

```

2220 GOSUB 3350
2230 GOTO 690
2240 CALL CLEAR
2250 PRINT "FREQUENZE TEORICHE"
2260 CALL MCHAR(24,3,145,28)
2270 PRINT ::
2280 PRINT "PI RAPPRESENTA LA PROBABILI-": "TA' CHE UNA MISURA CADA NEL-"
2290 PRINT "L'IESIMO INTERVALLO."
2300 PRINT "ALORA, PER M PROVE, M*PI E'": "LA FREQUENZA TEORICA PER"
2310 PRINT "OGNI INTERVALLO": ::
2320 GOSUB 3350
2330 CALL MCHAR(23,3,145,28)
2340 PRINT "N. - VALORE - PI - M*PI"
2350 CALL MCHAR(24,3,145,28)
2360 PRINT
2370 IF FF2=1 THEN 2390
2380 GOSUB 3440
2390 FOR I=0 TO N-1
2400 V=MIN+I*DX
2410 PI=EXP(-.5*((V-MEDIA)/SQM)^2)
2420 FI=PI*.39894*DX/SQM
2430 PRINT STR$(I+1);":TAB(5);V;TAB(14);INT(1000*PI)/1000;
2440 MP1=INT(10*(M*PI+.05))/10
2450 IF MP1=INT(MP1) THEN 2480
2460 MP=STR$(MP1)
2470 GOTO 2490
2480 MP=STR$(MP1)&".0"
2490 PRINT TAB(26-LEN(MP));MP;
2500 IF (I+1)/10<>INT((I+1)/10) THEN 2530
2510 GOSUB 3350
2520 CALL MCHAR(23,3,32,28)
2530 NEXT I
2540 CALL MCHAR(24,3,145,28)
2550 PRINT ::
2560 GOSUB 3350
2570 GOTO 690
2580 CALL CLEAR
2590 PRINT "CONFRONTO DEGLI ISTOGRAMMI"
2600 CALL MCHAR(24,3,145,28)
2610 PRINT "APPARIRANNO, A SINISTRA,": "L'ISTOGRAMMA DELLE FREQUENZE"
2620 PRINT "SPERIMENTALI E, A DESTRA,": "QUELLO DELLE FREQUENZE"
2630 PRINT "TEORICHE": ::
2640 GOSUB 3350
2650 CALL CLEAR
2660 IF N=15 THEN 2810
2670 IF N/2=INT(N/2) THEN 2750
2680 FOR I=0 TO N-3 STEP 2
2690 B(I/2)=(A(I)+A(I+1))/2
2700 NEXT I
2710 ULT=(N-1)/2
2720 B(ULT)=A(N-1)
2730 INIZ=INT((15-ULT)/2)+2
2740 GOTO 2860
2750 FOR I=0 TO N-2 STEP 2
2760 B(I/2)=(A(I)+A(I+1))/2
2770 NEXT I
2780 INIZ=INT((15-(N/2-1))/2)+2
2790 ULT=INT(N/2)-1
2800 GOTO 2860
2810 FOR I=0 TO N-1
2820 B(I)=A(I)
2830 NEXT I
2840 INIZ=INT((15-N)/2)+2
2850 ULT=N-1
2860 U=MAX/20
2870 COL=135
2880 GOSUB 3260
2890 IF FF2=1 THEN 2910
2900 GOSUB 3440
2910 IF N>15 THEN 2990
2920 FOR I=0 TO N-1
2930 V=MIN+I*DX
2940 MP1=EXP(-.5*((V-MEDIA)/SQM)^2)
2950 MP1=MP1*M*DX*.39894/SQM
2960 B(I)=MP1
2970 NEXT I
2980 GOTO 3210
2990 IF N/2=INT(N/2) THEN 3020
3000 NN=N-2
3010 GOTO 3030
3020 NN=N-1
3030 FOR I=0 TO NN
3040 V=MIN+I*DX
3050 IF I/2<>INT(I/2) THEN 3070
3060 MP1=0
3070 PI=EXP(-.5*((V-MEDIA)/SQM)^2)
3080 PI=PI*M*DX*.39894/SQM

```

```

3090 MP1=MPI+PI
3100 IF I/2=INT(I/2) THEN 3130
3110 IND=INT(I/2)
3120 B(IND)=MP1/2
3130 NEXT I
3140 IF N/2=INT(N/2) THEN 3190
3150 IND=INT(N/2)
3160 V=V+DX
3170 PI=EXP(-.5*((V-MEDIA)/SQM)^2)
3180 B(IND)=PI*M*DX*.39894/SQM
3190 INIZ=INT((15-N/2)/2)+17
3200 GOTO 3220
3210 INIZ=INT((15-N)/2)+17
3220 COL=143
3230 GOSUB 3260
3240 GOSUB 3350
3250 GOTO 690
3260 REM ***STAMPA***
3270 CALL HCHAR(22,2,144,30)
3280 FOR I=0 TO ULT
3290 Q=INT(B(I)/U)
3300 RESTO=B(I)-Q*U
3310 OTT=INT(RESTO*B/U)+1
3320 CALL VCHAR(22-Q,INIZ+1,COL,Q)
3330 IF OTT=0 THEN 3350
3340 CALL HCHAR(21-Q,INIZ+1,COL-B+OTT)
3350 NEXT I
3360 RETURN
3370 SOMMA=0
3380 FOR I=0 TO N-1
3390 SOMMA=SOMMA+(MIN+I*DX)*A(I)
3400 NEXT I
3410 MEDIA=SOMMA/M
3420 FF1=1
3430 RETURN
3440 IF FF1=1 THEN 3460
3450 GOSUB 3370
3460 SOMMA=0
3470 FOR I=0 TO N-1
3480 V=MIN+I*DX
3490 SCAR2=(V-MEDIA)^2
3500 SOMMA=SOMMA+SCAR2*A(I)
3510 NEXT I
3520 SQM=SQR(SOMMA/M)
3530 FF2=1
3540 RETURN
3550 PRINT TAB(6); "- PREMI UN TASTO -"
3560 CALL KEY(0,K,S)
3570 IF S=0 THEN 3560
3580 RETURN
3590 CALL CLEAR
3600 END

```

Alla riga 160 vengono dimensionate le variabili con indice che contengono i risultati sperimentali e teorici.

Le istruzioni DATA (linee 170-240) contengono le stringhe di caratteri esadecimali atte a definire i caratteri grafici per creare gli istogrammi. Nel confronto si è voluto che questi avessero colori differenti, così i caratteri sono stati inseriti contemporaneamente in due sottoinsiemi numerici distinti (linee 270-280).

L'input dei dati occupa tutte le linee da 450 a 690. M (650) è il totale delle misure; la variabile MAX (670) indica la massima frequenza ed è importante per il dimensionamento degli istogrammi.

Dopo il menù, alla riga 840, la variabile K rinvia alle varie sezioni del programma.

Le funzioni TAB nell'istruzione 930 servono ad incolonnare i valori numerici a destra, mentre la condizione seguente (linea 940) interrompe la scrittura della tabella ogni dieci righe. Il programma riprende alla pressione di un tasto.

La stampa degli istogrammi viene effettuata dalla routine 3260; prima di richiamarla è tuttavia necessario specificare alcune variabili, la qual cosa viene fatta alle righe 1090-1150. INIZ è la colonna dello schermo dalla quale inizia il grafico; quando l'istogramma è uno solo, esso è posto al centro mentre, quando si confrontano due istogrammi, ognuno viene centrato nella propria metà. U (linea 1130) è il valore unitario di un carattere-video (8 pixel); il grafico utilizza al massimo 20 righe, pertanto, se la massima frequenza da rappresentare è 60, ogni riga vale $U = 60/20 = 3$.

ULT indica alla routine quando è finita la stampa. Infine, COL è una variabile associata al colore; se COL = 135, l'istogramma sarà di colore blu, se COL = 127, il colore sarà rosso.

Il valore medio è calcolato dalla routine 3370 richiamata dalla GOSUB di riga 1240. La stampa (linea 1290) viene fatta con un arrotondamento a tre cifre decimali. Il passaggio alla routine 3370 mette a uno il flag FF1; se un'altra sezione del programma richiede il valore della media, controllando FF1 si saprà se è già stato calcolato, evitando così inutili ripetizioni.

Alla riga 1460, SCAR indica lo scarto dalla media; esso è stampato con due cifre decimali ma, prima di arrotondarlo, si aggiungono 5 millesimi così l'errore sarà inferiore.

Lo scarto quadratico medio è calcolato dalla routine 3440. Analogamente a quanto si è fatto in precedenza, il passaggio mette a uno il flag FF2. Lo scarto quadratico medio è contenuto nella variabile SQM; la sua stampa, a meno di un millesimo, si trova alla riga 1860. Il carattere 146, ivi presente, rappresenta la lettera greca μ , solitamente usata in questo caso.

Alle righe 1920-2230 viene calcolata la probabilità che una misura abbia a cadere in un dato intervallo indicato dall'utente. Il risultato si ottiene integrando la funzione della quale si è parlato nella premessa e che è rappresentata nella figura 19. La funzione, nel programma, è definita alla riga 2010 e l'integrazione, con il metodo di Simpson, occupa le righe seguenti fino alla 2160.

Alla riga 2420, PI indica le probabilità teoriche associate ad ogni risultato sperimentale. PI si trova calcolando la stessa precedente funzione $f(x)$ in corrispondenza dei risultati sperimentali e moltiplicando il valore ottenuto per la larghezza DX delle colonnine che formano l'istogramma. Le frequenze teoriche sono calcolate alla riga 2440.

Il confronto tra gli istogrammi occupa le righe 2860-3240.

La relativa complessità è dettata dal fatto che ogni singolo istogramma può essere formato anche da 30 colonne; nella maggior parte dei casi il TI 99/4A non potrebbe stampare i due grafici appaiati. Pertanto, se le colonne di ogni istogramma sono inferiori a 15 non sussistono problemi, ma se sono più di 15 viene attivato un algoritmo per sommare le fre-

quenze relative a due a due in modo da ridurre alla metà il numero delle colonne dei due grafici e permettere quindi la stampa contemporanea. La memoria richiesta è di 8,7 Kbyte.

La città degli esagoni

La città degli esagoni altro non è che la versione computerizzata della tavola di Galton o binostato. Una disposizione piramidale di esagoni genera una serie di corridoi lungo i quali scendono delle palline da un'unica via di accesso posta in alto. Ad ogni esagono la pallina ha $1/2$ di probabilità di andare a destra ed $1/2$ di andare a sinistra. Le palline vengono raccolte e contate alle uscite.

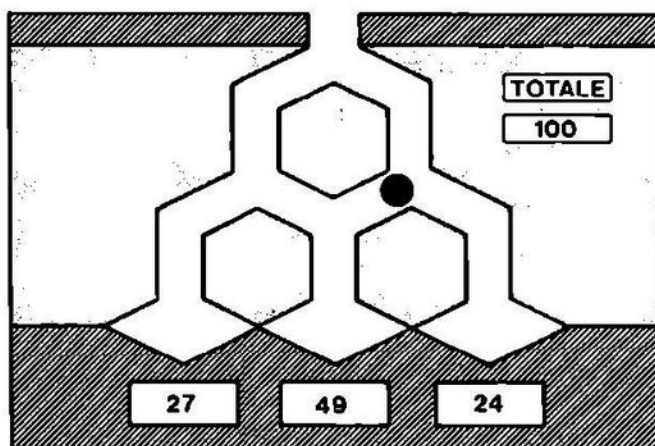
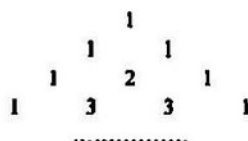


Fig. 20

È possibile scegliere da un minimo di due uscite ad un massimo di 4. Due uscite si ottengono ovviamente con un solo esagono ed i risultati sono analoghi a quelli che si ottengono con testa e croce relativamente al lancio di una moneta, le palline, cioè, stanno nel rapporto 1:1. Con tre vie di uscita le distribuzioni di probabilità sono $1/4$, $1/2$, $1/4$, ovvero le palline raccolte stanno nel rapporto di 1:2:1. Il risultato è assimilabile al lancio contemporaneo di due monete. Infine, con quattro vie di uscita, ogni pallina può scendere attraverso otto diversi percorsi e la distribuzione di probabilità è $1/8$, $3/8$, $3/8$, $1/8$; lanciando un gran numero di palline il loro numero alle uscite tenderà al rapporto 1:3:3:1. I numeri (1),

(1,1), (1,2,1), (1,3,3,1), ecc. sono i coefficienti binomiali altrimenti noti per costruire il triangolo di Tartaglia:



La distribuzione di probabilità appena esaminata viene detta binomiale (o bernoulliana).

Il programma è semplicemente una simulazione di ciò che è ottenibile con un apparecchio reale ed in quanto tale ha il pregio di essere esente da accidenti vari e da eventuali difetti di fabbricazione. L'unica difficoltà nella tecnica di programmazione, legata per altro ad un'esigenza puramente estetica, consiste nel creare il corretto movimento della pallina all'interno dei corridoi. Per questo motivo si è adottato l'Extended Basic, che, come è noto, è provvisto soprattutto di istruzioni CALL aggiuntive, rivolte alla grafica.

Lo sprite che rappresenta la pallina è fatto scendere con l'istruzione CALL MOTION, la sua posizione viene continuamente controllata con una CALL POSITION. Quando il numero di riga Y indica che lo sprite è in prossimità di un esagono, una funzione RND associata ad un'altra CALL MOTION, fa deviare la pallina a destra o a sinistra, diagonalmente, e dopo un altro valore di Y viene ripristinato il movimento lungo la verticale. Quando il percorso è più lungo, le istruzioni non garantiscono più il corretto movimento dello sprite ed è necessario introdurre qualche istruzione CALL LOCATE per correggere la posizione. Inoltre è da tener presente che mentre il Basic valuta la posizione e passa al successivo comando per modificare le componenti della velocità dello sprite, questo continua nel suo movimento tanto da far giungere in ritardo la successiva istruzione. La cosa è tanto più evidente quanto maggiore è la velocità. Il programmatore deve pertanto comportarsi come uno sciatore impegnato in uno slalom, il quale anticipa le porte. Se, supponiamo, la pallina deve deviare alla riga 58 (in pixel), l'istruzione dovrà essere IF Y<52... L'analogia con lo sciatore è evidente!

```

100 REM      LA CITTA'
110 REM      DEGLI ESAGONI
120 REM      *****
130 REM      di: SERGIO BORSANI
140 REM      versione:
150 REM      EXTENDED BASIC
160 REM      *****
170 CALL CHAR(128,"0000000030F3FFF")
180 CALL CHAR(129,"030F3FFFFFFFF")
190 CALL CHAR(130,"C0F0CFFFFFFF")
200 CALL CHAR(131,"0000000C0F0CFF")
210 CALL CHAR(132,"FF3F0F03")
220 CALL CHAR(133,"FFFFFFFF3F0F03")
```

```

230 CALL CHAR(134,"FFFFFFFFFFFFCFOC")
240 CALL CHAR(135,"FFCFC")
250 CALL CHAR(136,"FFFFFFFFFFFFFFFF"):: CALL CHAR(137,"O")
260 CALL CHAR(140,"071F3F7F7FFFFFFFFF7F7F3F1F07000080E0F0F8F8FCFCFCFC8F8F0E08")
270 CALL COLOR(13,9,1):: CALL COLOR(14,9,12)
280 CALL CHAR(120,"FFFFFFFFFFFFFFFF")
290 CALL CHAR(121,"C0F0CFFFFFFFFF")
300 CALL CHAR(122,"00000000C0F0CFF")
310 CALL CHAR(123,"0000000030F3FFF")
320 CALL CHAR(124,"030F3FFFFFFFFF")
330 CALL COLOR(12,7,12)
340 CALL CLEAR
350 RI=5 :: CO=15 :: GOSUB 2330
360 CALL HCHAR(6,8,136,6):: CALL HCHAR(7,8,136,6)
370 CALL HCHAR(6,20,136,6):: CALL HCHAR(7,20,136,6)
380 CALL VCHAR(8,6,136,10):: CALL VCHAR(8,25,136,10)
390 CALL HCHAR(18,8,136,18)
400 DISPLAY AT(11,11)SIZE(9):"LA CITTA'"
410 DISPLAY AT(13,13)SIZE(5):"DEGLI"
420 DISPLAY AT(15,12)SIZE(7):"ESAGONI"
430 FOR T=1 TO 500 :: NEXT T
440 CALL CLEAR
450 PRINT "VUOI LE SPIEGAZIONI? (S/N)"
460 CALL KEY(O,K,S)
470 IF S=0 THEN 460
480 IF K=B3 THEN 500
490 IF K=78 THEN 610 ELSE 460
500 CALL CLEAR
510 PRINT "UNA PALLINA VIENE FATTA SCENDERE ATTRAVERSO DEGLI ESAGO-"
520 PRINT "NI. AD OGNI BIVIO HA IL 50% DI PROBABILITA' DI ANDARE A"
530 PRINT "DESTRA O A SINISTRA.":: "IL COMPUTER SIMULA GLI EVEN-"
540 PRINT "TI E REGISTRA I RISULTATI.":: "PIU' O MENO NUMERO DI IL LO"
550 PRINT "RO NUMERO E' DETERMINATO DALLA SCELTA DELLE VIE DI USCITA"
560 PRINT "TA.":: "N"
570 PRINT TAB(6):"PREMI UN TASTO -"
580 CALL KEY(O,K,S)
590 IF S=0 THEN 590
600 CALL CLEAR
610 PRINT "QUANTE USCITE VUOI? (2-3-4)"
620 INPUT "U = ":U
630 PRINT
640 IF U<2 OR U>4 THEN 660 ELSE 670
650 PRINT "ERRORE: RIPROVA.":: "BOTTO 630"
660 PRINT "QUANTE PALLINE VUOI?"
670 INPUT "N = ":N
680 PRINT
690 IF N>1000 THEN PRINT "ESAGERATO! UN PO' MENO.":: "BOTTO 680"
700 CALL COLOR(13,9,12)
710 GOSUB 1950
720 ON U-1 BOTTO 740,1050,1450
730 DATA 1,120,10,11,121,1,12,122,1,13,137,2,13,123,1,14,124,1
740 DATA 1,121,1,18,122,1,19,137,2,21,123,1,22,124,1,23,120,10
750 RESTORE 740
760 FOR I=1 TO 12 :: READ C,W,V :: CALL HCHAR(8,C,W,V):: NEXT I
770 DATA 1,120,2,13,121,1,14,124,1,15,120,4,19,121,1,20,124,1,21,120,12
780 FOR I=1 TO 7 :: READ C,W,V :: CALL HCHAR(9,C,W,V):: NEXT I
800 CALL HCHAR(10,1,120,32)
810 DATA 1,11,16,2,22,1
820 FOR J=1 TO 3 :: RESTORE 810
830 FOR I=1 TO 3 :: READ C,V :: CALL HCHAR(10+J,C,120,V):: NEXT I
840 NEXT J
850 CALL HCHAR(14,1,120,32)
860 RANDOMIZE
870 FOR CNT=1 TO N
880 CALL SPRITE(81,140,5,1,122)
890 CALL MAGNIFY(3)
900 CALL MOTION(81,15,0)
910 CALL POSITION(81,Y,X)
920 IF Y<5 THEN 910
930 IF RND<5 THEN VEL=-32 ELSE VEL=32
940 CALL MOTION(81,15,VEL)
950 CALL POSITION(81,Y,X)
960 IF Y<20 THEN 950 ELSE CALL MOTION(81,25,0)
970 CALL POSITION(81,Y,X)
980 IF Y<49 THEN 970 ELSE CALL DELSPRITE(81)
990 IF X<128 THEN A(1)=A(1)+1 :: GOTO 1010
1000 A(2)=A(2)+1 :: DISPLAY AT(12,16)BIZE(4)BEEP:USING "###":A(2):: BOTTO 1020
1010 DISPLAY AT(12,10)BEEP SIZE(4):USING "###":A(1)
1020 GOSUB 1890
1030 NEXT CNT
1040 BOTTO 1920
1050 CALL HCHAR(13,1,120,7)
1060 R=13 :: FOR C=7 TO 19 STEP 6 :: GOSUB 2280 :: NEXT C
1070 CALL HCHAR(13,26,120,7)
1080 CALL HCHAR(14,1,120,64)
1090 FOR I=1 TO 3 :: CALL HCHAR(14,4+6*I,121)

```

```

1100 CALL HCHAR(14,5+6*1,124):: NEXT I
1110 DATA 1,8,13,2,19,2,25,8
1120 FOR I=1 TO 3:: RESTORE 1110
1130 FOR J=1 TO 4:: READ C,V:: CALL HCHAR(15+I,C,120,V)
1140 NEXT J:: NEXT I
1150 CALL HCHAR(19,1,120,32)
1160 RANDOMIZE
1170 FOR CNT=1 TO N
1180 CALL SPRITE(01,140,5,1,122,15,0)
1190 CALL MAGNIFY(3)
1200 CALL POSITION(01,Y,X)
1210 IF Y<6 THEN 1200
1220 IF RND<.5 THEN VEL=-34:: CO=98 ELSE VEL=34:: CO=146
1230 CALL MOTION(01,15,VEL)
1240 CALL POSITION(01,Y,X)
1250 IF Y<19 THEN 1240
1260 CALL MOTION(01,20,0)
1270 CALL POSITION(01,Y,X)
1280 IF Y<44 THEN 1270 ELSE CALL LOCATE(01,Y,CO)
1290 IF RND<.5 THEN VEL=-39 ELSE VEL=39
1300 CALL MOTION(01,15,VEL)
1310 CALL POSITION(01,Y,X)
1320 IF Y<58 THEN 1310 ELSE CALL MOTION(01,30,0)
1330 CALL POSITION(01,Y,X)
1340 IF Y<88 THEN 1330 ELSE CALL DELSPRITE(01)
1350 IF X>100 THEN 1380
1360 A(1)=A(1)+1:: DISPLAY AT(17,7)SIZE(4)BEEP:USING "###":A(1)
1370 GOTO 1420
1380 IF X>150 THEN 1410
1390 A(2)=A(2)+1:: DISPLAY AT(17,13)SIZE(4)BEEP:USING "###":A(2)
1400 GOTO 1420
1410 A(3)=A(3)+1:: DISPLAY AT(17,19)SIZE(4)BEEP:USING "###":A(3)
1420 GOSUB 1890
1430 NEXT CNT
1440 GOTO 1920
1450 CALL HCHAR(18,1,120,4)
1460 R=18:: FOR C=4 TO 22 STEP 6:: GOSUB 2280:: NEXT C
1470 CALL HCHAR(18,29,120,36)
1480 FOR I=1 TO 4:: CALL HCHAR(19,6+I+1,121)
1490 CALL HCHAR(19,6+I+2,124):: NEXT I
1500 DATA 1,5,10,2,16,2,22,2,28,5
1510 FOR I=1 TO 3:: RESTORE 1500
1520 FOR J=1 TO 5:: READ C,V:: CALL HCHAR(19+I,C,120,V)
1530 NEXT J:: NEXT I
1540 CALL HCHAR(23,1,120,32)
1550 RANDOMIZE
1560 FOR CNT=1 TO N
1570 CALL SPRITE(01,140,5,1,122,15,0)
1580 CALL MAGNIFY(3)
1590 CALL POSITION(01,Y,X)
1600 IF Y<5 THEN 1590
1610 IF RND<.5 THEN VEL=-32:: CO=98 ELSE VEL=32:: CO=146
1620 CALL MOTION(01,15,VEL)
1630 CALL POSITION(01,Y,X)
1640 IF Y<17 THEN 1630 ELSE CALL LOCATE(01,Y,CO)
1650 CALL MOTION(01,25,0)
1660 CALL POSITION(01,Y,X)
1670 IF Y<44 THEN 1660
1680 IF RND<.5 THEN VEL=-38:: CO=CO-24:: GOTO 1700
1690 VEL=38:: CO=CO+24
1700 CALL MOTION(01,15,VEL)
1710 CALL POSITION(01,Y,X)
1720 IF Y<55 THEN 1710
1730 CALL LOCATE(01,Y,CO):: CALL MOTION(01,35,0)
1740 CALL POSITION(01,Y,X)
1750 IF Y<82 THEN 1740
1760 IF RND<.5 THEN VEL=-38:: CO=CO-24:: GOTO 1780
1770 VEL=38:: CO=CO+24
1780 CALL MOTION(01,15,VEL)
1790 CALL POSITION(01,Y,X)
1800 IF Y<98 THEN 1790
1810 CALL LOCATE(01,Y,CO):: CALL MOTION(01,40,0)
1820 CALL POSITION(01,Y,X)
1830 IF Y<136 THEN 1820 ELSE CALL DELSPRITE(01)
1840 CO=(CO-2)/48:: A(CO)=A(CO)+1:: P=CO*6-2
1850 DISPLAY AT(21,P)SIZE(4)BEEP:USING "###":A(CO)
1860 GOSUB 1890
1870 NEXT CNT
1880 GOTO 1920
1890 DISPLAY AT(3,22)SIZE(6):"TOTALE"
1900 DISPLAY AT(5,22)SIZE(6):USING "#####":CNT
1910 RETURN
1920 DISPLAY AT(24,6)BEEP SIZE(18):"- PREMI UN TASTO -"
1930 CALL KEY(0,K,S)
1940 IF S=0 THEN 1930 ELSE 2420
1950 CALL CLEAR:: CALL HCHAR(1,1,120,32)

```

```

1960 CALL HCHAR(1,16,137,2)
1970 CALL HCHAR(2,1,137,192)
1980 DATA 2,1,136,13,2,14,134,1,2,15,133,1,2,18,132,1,2,19,133,1,2,20,136,13
1990 DATA 1,136,12,3,13,133,1,3,20,132,1,3,21,136,12
2000 RESTORE 1980
2010 FOR I=1 TO 10 : READ R,C,W,V : CALL HCHAR(R,C,W,V) : NEXT I
2020 FOR I=4 TO 6 : CALL HCHAR(1,1,136,12) : CALL HCHAR(1,21,136,12) : NEXT I
2030 DATA 1,136,10,11,134,1,12,135,1,21,132,1,22,133,1,23,136,10
2040 FOR I=1 TO 6
2050 READ C,W,V : CALL HCHAR(7,C,W,V) : NEXT I
2060 RI=4 : CO=15 : GOSUB 2330
2070 IF U=2 THEN 2270
2080 CALL HCHAR(8,1,137,160)
2090 DATA 1,136,9,10,133,1,23,132,1,24,136,9
2100 FOR I=1 TO 4 : READ C,W,V : CALL HCHAR(8,C,W,V) : NEXT I
2110 FOR I=9 TO 11 : CALL HCHAR(1,1,136,9) : CALL HCHAR(1,24,136,9) : NEXT I
2120 DATA 1,136,7,8,134,1,9,133,1,24,132,1,25,133,1,26,136,7
2130 FOR I=1 TO 6 : READ C,W,V : CALL HCHAR(12,C,W,V) : NEXT I
2140 RI=9 : CO=12 : GOSUB 2330
2150 RI=9 : CO=18 : GOSUB 2330
2160 IF U=3 THEN 2270
2170 CALL HCHAR(13,1,137,160)
2180 DATA 1,136,6,7,133,1,26,132,1,27,136,6
2190 FOR I=1 TO 4 : READ C,W,V : CALL HCHAR(13,C,W,V) : NEXT I
2200 FOR I=14 TO 16 : CALL HCHAR(1,1,136,6) : NEXT I
2210 CALL HCHAR(1,27,136,6) : NEXT I
2220 DATA 1,136,4,5,134,1,2,135,1,27,132,1,28,133,1,29,136,4
2230 FOR I=1 TO 6 : READ C,W,V : CALL HCHAR(17,C,W,V) : NEXT I
2240 RI=14 : CO=9 : GOSUB 2330
2250 RI=14 : CO=15 : GOSUB 2330
2260 RI=14 : CO=21 : GOSUB 2330
2270 RETURN
2280 DATA 121,122,137,137,123,124
2290 RESTORE 2280
2300 FOR I=1 TO 6 : READ W : CALL HCHAR(R,C+I,W) : NEXT I
2310 RETURN
2320 REM *** ESAGONO ***
2330 CALL HCHAR(RI,CO,128)
2340 CALL HCHAR(RI,CO+1,129)
2350 CALL HCHAR(RI,CO+2,130)
2360 CALL HCHAR(RI,CO+3,131)
2370 CALL HCHAR(RI+1,CO,136,4)
2380 CALL HCHAR(RI+2,CO,136,4)
2390 RI=RI+3
2400 FOR I=0 TO 3 : CALL HCHAR(RI,CO+I,132+I) : NEXT I
2410 RETURN
2420 CALL CLEAR
2430 PRINT "VUOI RICOMINCIARE? (S/N) : " : " : "
2440 CALL KEY(0,K,S)
2450 IF S=0 THEN 2440
2460 IF K=83 THEN 2480
2470 IF K=78 THEN 2490 ELSE 2440
2480 FOR I=1 TO 4 : A(I)=0 : NEXT I : GOTO 610
2490 CALL CLEAR : END

```

Il programma si apre con la definizione dei caratteri grafici; la lunga stringa di riga 280 definisce lo sprite con la forma di un cerchietto colorato, esso rappresenterà la pallina che scende tra gli esagoni. Le istruzioni 350 richiamano routine per disegnare un esagono dopo aver specificato le coordinate. Questa prima immagine appartiene al titolo. I dati in input sono: U, il numero delle vie di uscita e N, il numero delle palline. Si è stabilito che N non sia maggiore di 1000 ma nulla vieta di aumentare il limite con una lieve modifica del programma.

L'istruzione 730 provoca un salto condizionato a tre sezioni del programma ognuna con un diverso numero di esagoni. Data l'analoga struttura delle parti basterà esaminarne una a titolo di esempio. Scegliamo quella che va da riga 1050 a riga 1440. In essa tre esagoni sono disposti a triangolo (due alla base e uno al vertice); tre sono le vie di uscita. Le istruzioni 1050-1150 creano lo sfondo o meglio lo completano, in quanto gran parte di questo viene fatta con la GOSUB di riga 720.



L'istruzione FOR di riga 1170 fa ripetere l'intero ciclo tante volte quante sono le palline richieste.

Ogni pallina appare nella parte alta dello schermo con la CALL SPRITE di riga 1180 ed inizia subito la sua discesa con una velocità di 15 unità; come è noto le componenti della velocità possono variare in un intervallo che va da -127 a +127. Con l'istruzione 1200 si indaga subito sulla sua posizione, in particolare sul numero di riga espresso in pixel (Y). Se Y è inferiore a 6 si torna a chiedere la posizione altrimenti significa che la pallina è in prossimità del primo esagono e deve essere deviata lateralmente.

La deviazione è controllata da una funzione RND (riga 1220), precisamente se viene generato casualmente un numero inferiore a 0.5 la pallina andrà a sinistra (componente orizzontale della velocità uguale a -34) altrimenti verrà deviata a destra (velocità = 34). Il valore 34 è stato scelto perché, sommato ad una componente verticale di 15 unità, il moto risultante avviene lungo una direzione obliqua parallela ad un lato dell'esagono.

Sempre a riga 1220 si specifica anche l'esatto numero di colonna, in pixel, corrispondente al corridoio verticale lungo il quale dovrà scendere la pallina dopo aver percorso il tratto obliquo. Infatti non è possibile controllare esattamente il percorso della pallina solo con le istruzioni CALL POSITION e CALL MOTION ma è necessario ripristinare periodicamente l'esatta posizione con un'istruzione CALL LOCATE. È quanto viene fatto a riga 1280 se la pallina è scesa sotto quota 44.

Le righe 1290-1300 provocano l'ultima deviazione mentre, sotto quota 58, riprende il moto verticale che si arresta al livello 88 dove la pallina scompare.

In base al numero di colonna X, restituito dall'ultima CALL POSITION, si deduce da quale via è uscita la pallina e si può incrementare il contatore corrispondente, A(1), A(2) oppure A(3) (righe 1350-1410). La GO-SUB di linea 1420 richiama il brevissimo sottoprogramma che incrementa il contatore generale CNT.

Il ciclo riprende.

Fisica

Le forze

Poiché le forze sono grandezze vettoriali, il programma è preceduto da alcuni elementi di calcolo vettoriale (sezioni 1 e 2).

La somma di due vettori è un vettore di intensità

$$V = \sqrt{V_1^2 + V_2^2 + 2V_1V_2 \cos \theta}$$

dove V_1 e V_2 sono le intensità dei vettori da sommare e θ è l'angolo compreso tra i vettori $\vec{V}_1$ e $\vec{V}_2$.

L'angolo α formato dal vettore somma $\vec{V}$ e $\vec{V}_1$ si ricava con la formula

$$\frac{V}{\sin \theta} = \frac{V_2}{\sin \alpha}$$

dalla quale si ottiene: $\sin \alpha = \sin \theta V_2/V$

In Basic non esiste la funzione arcsin che restituisce l'ampiezza dell'angolo, pertanto essa va definita come segue: $\text{ARCSIN} = \text{ATN}(X/\text{SQR}(1-X^2))$, dove X è il valore di $\sin \alpha$.

Gli argomenti delle funzioni trigonometriche vanno espressi in radianti; per ottenere il valore in radianti si moltiplicano i gradi per $\pi/180 = 0.0174532925\dots$ e per convertire in gradi il corrispondente valore in radianti, questo va moltiplicato per $180/\pi = 57.295779513\dots$

Un vettore può essere scomposto in due componenti ortogonali: $\vec{V} =$

$\bar{u}_x V_x + \bar{u}_y V_y$, dove $\bar{u}_x$ e $\bar{u}_y$ sono vettori unitari o versori, e $V_x = V \cos \alpha$ e $V_y = V \sin \alpha$.

Il programma inoltre esegue le seguenti operazioni: composizione di forze concorrenti, composizione di forze parallele, momento di una forza, equilibrio di un corpo rigido.

Sono forze concorrenti quelle applicate ad un medesimo punto. Vengono accettate fino ad un massimo di dieci forze, quindi non è necessario il dimensionamento delle matrici (o variabili con indice). Ogni forza viene scomposta nelle componenti ortogonali parallele agli assi cartesiani. Dette R_x ed R_y le componenti della risultante, risulta:

$$R_x = \sum F_{ix} \quad \text{e} \quad R_y = \sum F_{iy}$$

Il modulo della forza risultante si trova applicando il teorema di Pitagora, $R = \sqrt{R_x^2 + R_y^2}$. L'angolo, in radianti, formato dalla forza risultante con l'asse delle x è dato dalla funzione ATN (arcotangente) il cui argomento è $R_y/R_x = \tan \alpha / \cos \alpha = \tan \alpha$. Prima di far calcolare al computer R_y/R_x , bisogna accertarsi che R_x sia diverso da zero altrimenti compare un messaggio di errore. Naturalmente questa accortezza spetta al programmatore il quale deve prevedere anche questi risultati particolari ed inserire le opportune istruzioni.

Per la somma di forze parallele (anche qui fino ad un massimo di dieci) valgono le seguenti relazioni:

$$R = \sum_i F_i \quad \text{e} \quad \tau = \sum_i \tau_i$$

cioè, la risultante (R) ha un modulo uguale alla somma di tutte le intensità delle forze ed il momento della risultante (τ) è uguale alla somma di tutti i momenti. Se le forze sono applicate tutte ad una stessa retta, il punto di applicazione della risultante è:

$$x = \frac{\sum F_i x_i}{\sum F_i}$$

dove x_i sono le distanze dei punti di applicazione delle forze i -esime. Non è importante dove viene posta l'origine rispetto alla quale vengono calcolate queste distanze, tuttavia, le distanze poste alla destra dell'origine vanno prese col segno più, mentre quelle alla sinistra dell'origine vanno prese col segno meno. Parimenti si può convenire che le intensità delle forze abbiano senso positivo se le forze sono rivolte verso l'alto e segno negativo se sono rivolte verso il basso.

Quando una forza viene applicata ad un corpo che può ruotare attorno ad un punto O , l'esperienza stessa ci dice che essa è tanto più efficace quanto maggiore è la distanza tra la direzione della forza ed il punto O ,

ovvero quanto maggiore è il braccio della forza. Viene definito momento di una forza il prodotto della forza per il suo braccio. Per semplicità non si è tenuto conto del carattere vettoriale del momento.

Detta r la distanza del punto di applicazione della forza dal punto O , risulta: $b = r \sin \theta$ e $\tau = Fr \sin \theta$, dove θ è l'angolo formato dalle direzioni di F ed r .

Ogni sezione del programma è dotata di illustrazioni che rendono più chiare le relazioni tra i vari elementi ai quali ci si riferisce e rendono più agevole l'input dei dati all'utente.

L'ultima parte tratta l'equilibrio di un corpo rigido.

In ogni situazione valgono le seguenti regole: (1) la somma di tutte le forze deve essere uguale a zero; (2) la somma dei momenti di tutte le forze deve essere uguale a zero.

Non è possibile creare un programma in grado di risolvere le innumerevoli situazioni che rientrano in quella branca della fisica chiamata statica. Un solo problema viene trattato in particolare: l'equilibrio di una trave che appoggia su due punti A e B . Specificando le forze agenti sulla trave ed i loro punti di applicazione, vengono calcolate le reazioni sulla trave da parte dei vincoli A e B . Il metodo consiste nell'uguagliare a zero la somma di tutti i momenti delle forze rispetto al punto A ottenendo un'equazione di primo grado dove l'unica incognita è la forza nel punto B . Uguagliando a zero la somma di tutte le forze e sostituendo alla forza nel punto B il valore appena calcolato, si ottiene una nuova equazione di primo grado ad una incognita la cui soluzione è la forza nel punto A .

Prima di passare ad alcune applicazioni esaminiamo brevemente il listato.

```

100 REM          LE FORZE
110 REM *****
120 REM DI: SERGIO BORSANI
130 REM VERSIONE: TI BASIC
140 CALL CLEAR
150 CALL CHAR(128,"F")
160 CALL COLOR(13,5,5)
170 PRINT TAB(11); "LE FORZE":
180 CALL MCHAR(10,11,128,12)
190 CALL VCHAR(11,11,128,3)
200 CALL VCHAR(11,22,128,3)
210 CALL MCHAR(14,11,128,12)
220 CALL CHAR(135,"0102040B10204080")
230 CALL CHAR(137,"070E32C1010101FF")
240 CALL CHAR(138,"B0404020202020FF")
250 CALL CHAR(139,"00000000000000FF")
260 CALL CHAR(140,"00000000000000FF")
270 CALL CHAR(141,"0204")
280 CALL CHAR(142,"0102040B10204080")
290 CALL CHAR(143,"000000000030C30C0")
300 CALL CHAR(144,"030C30C")
310 CALL CHAR(145,"0F03050910204080")
320 CALL CHAR(146,"0F0E34CB10204080")
330 CALL CHAR(147,"0102040B10304884")
340 CALL CHAR(148,"0102040B102040FF")
350 CALL CHAR(149,"0101010101010101")
360 CALL CHAR(150,"0103050101010101")
370 CALL CHAR(151,"00B04")
380 CALL CHAR(152,"B0B0B0B0B0B0B0B0")
390 CALL CHAR(153,"00304B4B3")
400 CALL CHAR(154,"0F0E34C")

```

```

410 CALL CHAR(155,"000000000030C30FF")
420 CALL CHAR(156,"000000000032C70F0")
430 CALL CHAR(157,"01010101010101FF")
440 CALL CHAR(158,"0808080808080808")
450 CALL CHAR(159,"08080808082A1C08")
460 CALL CHAR(123,"0103070F1F7FFF")
470 CALL CHAR(124,"0080C0E0F0F8FCFE")
480 CALL CLEAR
490 PRINT "PER LA TUA SCELTA PREMI:", "IL NUMERO CORRISPONDENTE":
500 PRINT "1. SOMMA DI DUE VETTORI":
510 PRINT "2. COMPONENTI DI UN VETTORE":
520 PRINT "3. COMPOSIZIONE DI FORZE": "CONCORRENTI":
530 PRINT "4. COMPOSIZIONE DI FORZE": "PARALLELE":
540 PRINT "5. MOMENTO DI UNA FORZA":
550 PRINT "6. EQUILIBRIO DI UN CORPO": "RIGIDO":
560 PRINT "7. FINE.":
570 CALL KEY(0,K,S)
580 IF B=0 THEN 570
590 K=K-48
600 IF (K<1)+(K>7) THEN 570
610 ON K GOTO 620,1270,1660,2390,2940,3340,4010
620 CALL CLEAR
630 PRINT "SOMMA DI DUE VETTORI":
640 GOSUB 3980
650 PRINT "V=V1+V2"
660 CALL HCHAR(22,11,139,5)
670 CALL HCHAR(22,16,140)
680 CALL HCHAR(23,16,141)
690 CALL HCHAR(22,11,148)
700 FOR I=1 TO 4
710 CALL HCHAR(22-I,11+I,142)
720 NEXT I
730 CALL HCHAR(17,16,145)
740 FOR I=1 TO 6
750 CALL HCHAR(23-I,16+I,142)
760 NEXT I
770 CALL HCHAR(16,17,139,6)
780 CALL HCHAR(22,11,136)
790 CALL HCHAR(21,12,147)
800 CALL HCHAR(22,12,137)
810 FOR I=1 TO 4
820 CALL HCHAR(22-I,11+2*I,143)
830 CALL HCHAR(22-I,12+2*I,144)
840 NEXT I
850 CALL HCHAR(17,21,143)
860 CALL HCHAR(17,22,146)
870 CALL HCHAR(22,13,138)
880 DATA 23,14,86,23,15,49,18,13,86,18,14,30,18,18,86
890 RESTORE 880
900 FOR I=1 TO 3
910 READ RIGA,COLA,NUME
920 CALL HCHAR(RIGA,COLA,NUME)
930 NEXT I
940 PRINT
950 PRINT "SCRIVI I MODULI DEI VETTORI":
960 FOR I=1 TO 2
970 PRINT TAB(6); "V";STR$(I); " = ";
980 INPUT V(I)
990 NEXT I
1000 PRINT
1010 PRINT "SCRIVI L'ANGOLO FORMATO DAI": "DUE VETTORI"
1020 INPUT " (GRADI) ":A
1030 PRINT
1040 A=A*.0174532925
1050 R=V(1)*V(1)+V(2)*V(2)
1060 R=R+2*V(1)*V(2)*COS(A)
1070 R=SOR(R)
1080 X=BIN(A)*V(2)/R
1090 ARCSIN=ATN(X/SQR(1-X*X))
1100 A=ARCSIN*57.2957796
1110 GRADI=INT(A)
1120 A=(A-INT(A))*60
1130 PRIMI=INT(A)
1140 A=(A-INT(A))*60
1150 SECONDI=INT(A)
1160 PRINT
1170 PRINT "IL MODULO DEL VETTORE SOMMA": "E'":
1180 PRINT TAB(8); "V =";INT(1000*R)/1000:
1190 PRINT "L'ANGOLO FORMATO DA V E V1": "E'":
1200 PRINT TAB(8); "":
1210 PRINT TAB(8); "A = ";STR$(GRADI);CHR$(153);
1220 IF PRIMI=0 THEN 1240
1230 PRINT STR$(PRIMI);CHR$(39);STR$(SECONDI);CHR$(34)
1240 PRINT :
1250 GOSUB 3980
1260 GOTO 4010
1270 CALL CLEAR

```

```

1280 PRINT "COMPONENTI DI UN VETTORE":
1290 GOSUB 3980
1300 PRINT "::::::::::"
1310 FOR I=1 TO 6
1320 CALL HCHAR(22-I,9+2*I,143)
1330 CALL HCHAR(22-I,10+2*I,144)
1340 NEXT I
1350 CALL HCHAR(16,22,154)
1360 CALL HCHAR(21,11,155)
1370 CALL HCHAR(21,12,137)
1380 CALL HCHAR(21,13,139,9)
1390 CALL HCHAR(21,22,140)
1400 CALL HCHAR(22,22,141)
1410 CALL VCHAR(16,10,149,6)
1420 CALL HCHAR(16,10,150)
1430 CALL HCHAR(16,11,151)
1440 CALL VCHAR(16,23,152,6)
1450 CALL HCHAR(15,11,139,12)
1460 DATA 17,17,86,17,11,86,17,12,121,22,19,86,22,20,120
1470 RESTORE 1460
1480 FOR I=1 TO 5
1490 READ RIGA,COLO,NUME
1500 CALL HCHAR(RIGA,COLO,NUME)
1510 NEXT I
1520 PRINT "SCRIVI IL MODULO DEL VETTORE":
1530 INPUT "V = ":VV
1540 PRINT
1550 PRINT "SCRIVI IL VALORE DELL'ANGOLO": "IN GRADI": ""
1560 INPUT "A = ":A
1570 PRINT
1580 A=A*.0174532925
1590 V1=INT(100*(VV*COS(A)+.005))/100
1600 V2=INT(100*(VV*SIN(A)+.005))/100
1610 PRINT "LE COMPONENTI DEL VETTORE": "SONO":
1620 PRINT "Vx =":V1:
1630 PRINT "Vy =":V2:
1640 GOSUB 3980
1650 GOTO 4010
1660 CALL CLEAR
1670 PRINT "COMPOSIZIONE DI ": "FORZE CONCORRENTI":
1680 GOSUB 3980
1690 PRINT "::::::::::"
1700 FOR I=1 TO 5
1710 CALL HCHAR(19-I,12+I,142)
1720 NEXT I
1730 CALL HCHAR(14,17,145)
1740 CALL HCHAR(18,13,148)
1750 CALL HCHAR(18,14,139,10)
1760 CALL HCHAR(18,19,140)
1770 CALL HCHAR(19,19,141)
1780 FOR I=1 TO 3
1790 CALL HCHAR(18+I,14-2*I,144)
1800 CALL HCHAR(18+I,13-2*I,143)
1810 NEXT I
1820 CALL HCHAR(21,7,156)
1830 CALL VCHAR(13,15,149,6)
1840 DATA 19,22,88,13,13,89,19,17,70,19,18,49,15,17,70
1850 DATA 15,18,50,20,7,70,20,8,51
1860 RESTORE 1840
1870 FOR I=1 TO 8
1880 READ RIGA,COLO,NUME
1890 CALL HCHAR(RIGA,COLO,NUME)
1900 NEXT I
1910 INPUT "QUANTI SONO I VETTORI? ":N
1920 PRINT
1930 IF N<=10 THEN 1960
1940 PRINT "SPIACENTE, NON PIU' DI 10.":
1950 GOTO 1910
1960 PRINT "SCRIVI I MODULI DEI VETTORI":
1970 FOR I=1 TO N
1980 PRINT "F":STR$(I); " = ":
1990 INPUT V(I)
2000 NEXT I
2010 PRINT
2020 PRINT "... ED I RISPETTIVI ANGOLI": "FORMATI CON L'ASSE DELLE X,"
2030 PRINT "IN GRADI.":
2040 FX=0
2050 FY=0
2060 FOR I=1 TO N
2070 PRINT "A":STR$(I); " = ":
2080 INPUT A
2090 A=A*.0174532925
2100 FX=FX+V(I)*COS(A)
2110 FY=FY+V(I)*SIN(A)
2120 NEXT I
2130 PRINT :

```

```

2140 F=SQR(FX*FX+FY*FY)
2150 F=INT(100*(F+.005))/100
2160 IF FX=0 THEN 2200
2170 X=FY/FX
2180 X=INT(10*ATN(X)*37.29578)/10
2190 GOTO 2210
2200 X=90
2210 FX=INT(100*(FX+.005))/100
2220 FY=INT(100*(FY+.005))/100
2230 PRINT "LE COMPONENTI DELLA FORZA": "RISULTANTE SONO:"
2240 PRINT "FX =" : FX
2250 PRINT "FY =" : FY
2260 PRINT "IL MODULO DELLA FORZA RISUL-" : "TANTE E'":
2270 PRINT "F =" : F
2280 PRINT "L'ANGOLO FORMATO CON L'ASSE": "DELLE X E'":
2290 IF FX<>0 THEN 2340
2300 IF FY<0 THEN 2320
2310 GOTO 2360
2320 X=-X
2330 GOTO 2360
2340 IF FX>0 THEN 2360
2350 X=X+180
2360 PRINT "A =" : STR$(X);CHR$(153);
2370 GOSUB 3980
2380 GOTO 4010
2390 CALL CLEAR
2400 PRINT "COMPOSIZIONE": "DI FORZE PARALLELE":
2410 GOSUB 3980
2420 PRINT "":
2430 CALL HCHAR(18,12,139,11)
2440 CALL HCHAR(18,23,140)
2450 CALL HCHAR(19,23,141)
2460 CALL VCHAR(14,11,149,5)
2470 CALL HCHAR(13,11,150)
2480 CALL HCHAR(13,12,151)
2490 CALL VCHAR(19,13,158,3)
2500 CALL HCHAR(22,13,159)
2510 CALL VCHAR(15,16,149,3)
2520 CALL HCHAR(18,16,157)
2530 CALL HCHAR(14,16,150)
2540 CALL HCHAR(14,17,151)
2550 CALL VCHAR(19,19,158,2)
2560 CALL HCHAR(21,19,159)
2570 DATA 19,24,88,13,10,89,19,11,79,21,14,70,21,15,49
2580 DATA 15,14,70,15,15,50,20,20,70,20,21,51
2590 RESTORE 2570
2600 FOR I=1 TO 9
2610 READ RIGA,COLA,NUME
2620 CALL HCHAR(RIGA,COLA,NUME)
2630 NEXT I
2640 INPUT "QUANTE SONO LE FORZE? ":N
2650 PRINT
2660 IF N<10 THEN 2690
2670 PRINT "SPIACENTE, NON PIU' DI 10.":
2680 GOTO 2640
2690 PRINT "SCRIVI L'INTENSITA' DELLE": "FORZE E LE DISTANZE DEI RI-"
2700 PRINT "SPETTIVI PUNTI DI APPLICA-": "ZIONE DALL'ORIGINE":
2710 F=0
2720 T=0
2730 FOR I=1 TO N
2740 PRINT "F";STR$(I); " = "
2750 INPUT X
2760 PRINT "DISTANZA = "
2770 INPUT A
2780 F=F+X
2790 T=T+X*A
2800 PRINT
2810 NEXT I
2820 PRINT "LA RISULTANTE DELLE FORZE E'":
2830 IF F<>0 THEN 2860
2840 PRINT TAB(6); "F = 0":
2850 GOTO 2920
2860 F=INT(100*(F+.005))/100
2870 X=INT(100*(T/F+.005))/100
2880 PRINT TAB(6); "F =" : F
2890 PRINT "IL SUO PUNTO DI APPLICAZIONE"
2900 PRINT "E' POSTO AD UNA DISTANZA": "DALL'ORIGINE":
2910 PRINT TAB(6); "X =" : X
2920 GOSUB 3980
2930 GOTO 4010
2940 CALL CLEAR
2950 PRINT "MOMENTO DI UNA FORZA":
2960 GOSUB 3980
2970 PRINT "":
2980 FOR I=1 TO 6
2990 CALL HCHAR(22-1,11+2*I,143)
3000 CALL HCHAR(22-1,12+2*I,144)

```

```

3010 NEXT I
3020 CALL HCHAR(18,20,134)
3030 CALL HCHAR(17,13,139,8)
3040 CALL HCHAR(17,21,155)
3050 CALL HCHAR(17,22,137)
3060 CALL HCHAR(17,23,139,2)
3070 CALL HCHAR(17,24,140)
3080 CALL HCHAR(18,25,141)
3090 CALL VCHAR(18,12,149,5)
3100 DATA 22,12,79,20,17,82,18,24,70,19,11,66,18,21,65
3110 RESTORE 3100
3120 FOR I=1 TO 5
3130 READ RIGA,COLO,NUME
3140 CALL HCHAR(RIGA,COLO,NUME)
3150 NEXT I
3160 PRINT "SCRIVI LA LUNGHEZZA DI R.":
3170 INPUT "R = ":X
3180 PRINT "SCRIVI L'INTENSITA' DELLA FORZA F.":
3190 INPUT "F = ":F
3200 PRINT "SCRIVI L'ANGOLO FORMATO DAI":
3210 PRINT "DUE VETTORI: A ED F, IN GRADI":
3220 INPUT "A = ":A
3230 A=A*.01745329252
3240 T=X*F*SIN(A)
3250 T=INT(100*T)/100
3260 X=INT(100*X*SIN(A))/100
3270 PRINT "":
3280 PRINT "IL BRACCIO DELLA FORZA E'":
3290 PRINT TAB(6);"B =":X:
3300 PRINT "IL MOMENTO DELLA FORZA E'":
3310 PRINT TAB(6);"T =":T:
3320 GOSUB 3980
3330 GOTO 4010
3340 CALL CLEAR
3350 PRINT "EQUILIBRIO": "DI UN CORPO RIGIDO":
3360 GOSUB 3980
3370 PRINT "UN CORPO E' IN EQUILIBRIO": "QUANDO SONO SODDISFATTE LE"
3380 PRINT "SEGUENTI CONDIZIONI":
3390 PRINT "1. LA SOMMA DELLE FORZE DEVE": "ESSERE UGUALE A ZERO.":
3400 PRINT "2. LA SOMMA DEI MOMENTI DI": "TUTTE LE FORZE DEVE ESSE-":
3410 PRINT "RE UGUALE A ZERO.":
3420 PRINT TAB(6);"PREMI UN TASTO -"
3430 CALL KEY(0,K,S)
3440 IF S=0 THEN 3430
3450 CALL CLEAR
3460 PRINT "UNA TRAVE E' IN EQUILIBRIO": "SUI PUNTI A E B ED E' SOG-":
3470 PRINT "GETTA AD N FORZE":
3480 PRINT "IL PROGRAMMA PERMETTE DI": "CALCOLARE LE FORZE ESERCITA-":
3490 PRINT "TE SULLA SBARRA IN A E IN B.":
3500 CALL HCHAR(16,11,128,12)
3510 CALL HCHAR(17,13,123)
3520 CALL HCHAR(17,14,124)
3530 CALL HCHAR(17,19,123)
3540 CALL HCHAR(17,20,124)
3550 CALL VCHAR(17,11,158,2)
3560 CALL HCHAR(19,11,159)
3570 CALL VCHAR(17,16,158,4)
3580 CALL HCHAR(21,16,159)
3590 CALL VCHAR(17,22,158,3)
3600 CALL HCHAR(20,22,159)
3610 CALL HCHAR(19,13,65)
3620 CALL HCHAR(19,19,66)
3630 INPUT "QUANTE SONO LE FORZE? ":N
3640 PRINT
3650 IF N<=10 THEN 3680
3660 PRINT "SPIACENTE, NON PIU' DI 10.":
3670 GOTO 3630
3680 PRINT "PER OGNI FORZA SCRIVI L'IN-": "TENSITA' E LA DISTANZA DAL"
3690 PRINT "PUNTO A.":
3700 F=0
3710 T=0
3720 FOR I=1 TO N
3730 PRINT TAB(6);"F";STR$(I);" = ":
3740 INPUT F
3750 PRINT TAB(6);"X";STR$(I);" = ":
3760 INPUT X
3770 F=F+A
3780 T=T+A*X
3790 PRINT
3800 NEXT I
3810 PRINT "ORA SCRIVI LA DISTANZA AB":
3820 INPUT "AB = ":X
3830 IF X<0 THEN 3860
3840 PRINT "SPIACENTE, DEVE ESSERE X<0.":
3850 GOTO 3820
3860 V(2)=INT(100*(-T/X+.005))/100

```

```

3870 V(1)=-F-V(2)
3880 PRINT :
3890 PRINT "LE FORZE ESERCITATE SULLA"
3900 PRINT "TRAVE NEI PUNTI A E B SONO:"
3910 FOR I=1 TO 2
3920 PRINT TAB(6); "F"(CHR$(96+I)); " = "V(I)
3930 PRINT
3940 NEXT I
3950 PRINT :
3960 GOSUB 3980
3970 GOTO 4010
3980 CALL HCHAR(24,3,128,28)
3990 PRINT :
4000 RETURN
4010 PRINT "VUOI CONTINUARE? (S/N) "
4020 CALL KEY(0,K,S)
4030 IF S=0 THEN 4020
4040 IF K=B3 THEN 480
4050 IF K=78 THEN 4070
4060 GOTO 4020
4070 CALL CLEAR
4080 END

```

Dopo il titolo (linea 170) compaiono le definizioni dei caratteri grafici; ogni sezione del programma è accompagnata da una figura che illustra la situazione che si sta trattando rendendo più chiara, allo stesso tempo, la comprensione di quali dati vadano introdotti. Diciamo subito che le figure sono statiche e non rispecchiano fedelmente il problema che si sta affrontando né cambiano al variare dei dati in input.

La tecnica di programmazione non presenta difficoltà; per i calcoli si seguono integralmente le formule riportate nella parte iniziale. Le varie sezioni presentano la stessa struttura: sottotitolo, parte grafica-illustrativa, input dei dati, calcolo, output dei risultati.

Alla riga 989 V(I) contiene il modulo dei vettori dei quali si chiede la somma; A è l'angolo in gradi. Alla riga 1040 l'angolo viene trasformato in radianti mentre le istruzioni seguenti (1050-1070) calcolano il modulo della somma. Alla riga 1080 si applica il teorema dei seni e alla 1100 l'angolo viene trasformato in gradi decimali e successivamente (righe 1110-1150) in gradi sessagesimali. Prima della stampa (riga 1230) i valori numerici sono trasformati in stringhe poiché altrimenti rimarrebbe uno spazio vuoto tra i numeri e i simboli che rappresentano i gradi, i primi e i secondi.

Alla riga 1990, i vettori da sommare vengono associati a V(I). Ad ogni input dell'angolo A (linea 2080) si calcolano le componenti dei vettori e contemporaneamente la loro somma; al termine, FX è la somma delle componenti parallele all'asse X ed FY è la somma delle componenti parallele all'asse Y (righe 2100-2110).

La variabile F di linea 2150 contiene il modulo della risultante mentre la X di riga 2180 è l'angolo che la risultante forma con l'asse delle X. Non bisogna dimenticare che la funzione arcotangente (ATN) restituisce un valore che, trasformato in gradi, è compreso tra -90° e $+90^\circ$, estremi esclusi; in base alle componenti FX ed FY si può tuttavia provvedere a dare il risultato con valori dell'angolo compresi tra 0° e 360° .

La quarta sezione calcola la risultante di forze parallele. Alle linee 2710-

2720 F e T sono rispettivamente la somma delle forze e la somma dei momenti delle forze. In questo caso la variabile A (input di riga 2770) non rappresenta un angolo ma la distanza del punto di applicazione; è preferibile non riservare altre aree di memoria con nuovi nomi di variabili!

Il programma occupa 11 Kbyte.

Ci trasferiamo ora alla tastiera per due esempi di applicazioni.

1. Si voglia calcolare la risultante delle forze rappresentate in figura 21.

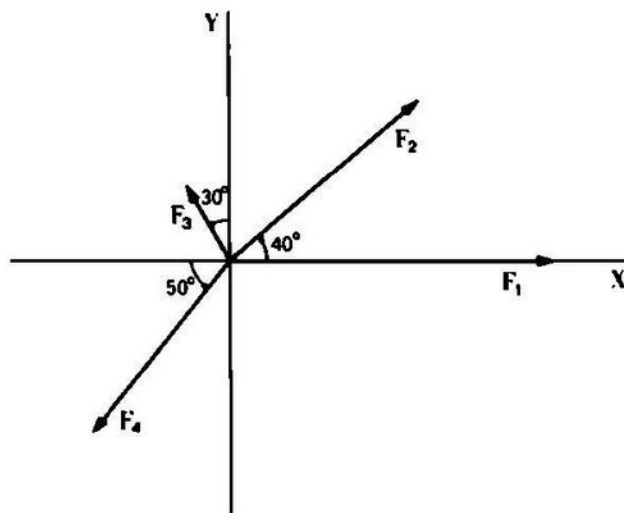
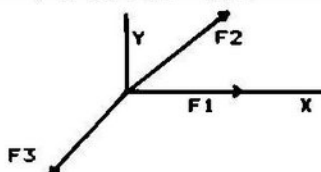


Fig. 21

Le loro intensità sono $F_1=1200$ kgf, $F_2=900$ kgf, $F_3=300$ kgf, ed $F_4=800$ kgf. Gli angoli vanno riferiti al verso positivo dell'asse delle x . Vediamo come vengono introdotti i dati e le risposte del computer.

RUN

COMPOSIZIONE DI FORZE CONCORRENTI



QUANTI SONO I VETTORI? 4 <ENTER>

SCRIVI I MODULI DEI VETTORI

F1 = ? 1200 <ENTER>

F2 = ? 900 <ENTER>

F3 = ? 300 <ENTER>

F4 = ? 800 <ENTER>

... ED I RISPETTIVI ANGOLI
FORMATI CON L'ASSE DELLE X,
IN GRADI.

A1 = ? 0 <ENTER>

A2 = ? 40 <ENTER>

A3 = ? 120 <ENTER>

A4 = ? 230 <ENTER>

LE COMPONENTI DELLA FORZA
RISULTANTE SONO:

FX = 1225.21

FY = 225.48

IL MODULO DELLA FORZA RISUL-
TANTE E':

F = 1245.79

L'ANGOLO FORMATO CON L'ASSE
DELLE X E':

A = 10.4°

VUOI CONTINUARE? (S/N)

2. Una trave appoggia su due punti *A* e *B*. Determinare le reazioni sui vincoli conoscendo le forze applicate (vedi figura 22).

Bisogna innanzitutto stabilire un verso positivo per le forze. Se F_1 , F_2 , F_3 , ecc. vengono introdotte col segno positivo, F_x e F_y risultano negative. Se eccezionalmente una delle due dovesse risultare positiva significa che la trave non può restare in equilibrio stando semplicemente appoggiata ma deve essere in qualche modo trattenuta.

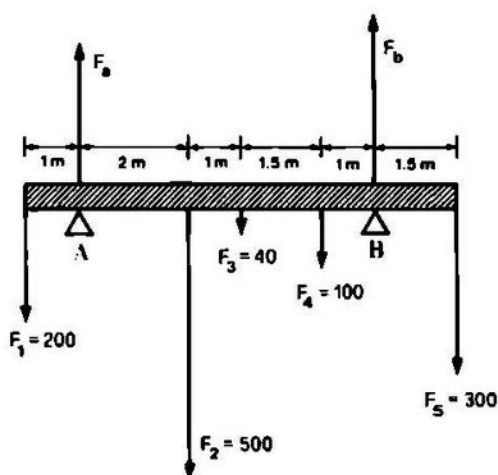


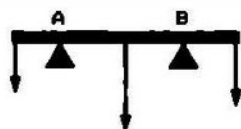
Fig. 22

Le distanze dei punti di applicazione devono essere riferite al punto A; punti alla destra di A hanno distanze positive, quelli alla sinistra distanze negative.

Ecco come appare l'esecuzione dell'ultima parte del programma.

RUN

**EQUILIBRIO
DI UN CORPO RIGIDO**



QUANTE SONO LE FORZE? 5

<ENTER>

PER OGNI FORZA SCRIVI L'IN-
TENSITA' E LA DISTANZA DAL
PUNTO A.

F1 = ? 200
X1 = ? -1

<ENTER>
<ENTER>

F2 = ? 500
X2 = ? 2

<ENTER>
<ENTER>

F3 = ? 40
X3 = ? 3

<ENTER>
<ENTER>

F4 = ? 100
X4 = ? 4.5

<ENTER>
<ENTER>

```

FS = ? 300          <ENTER>
XS = ? 7            <ENTER>

ORA SCRIVI LA DISTANZA AB
AB = 5.5            <ENTER>

LE FORZE ESERCITATE SULLA
TRAVE NEI PUNTI A E B SONO:

Fa = -509.09
Fb = -630.91

VUOI CONTINUARE? (S/N)

```

Cinematica

Il programma affronta i due principali tipi di moto: il moto rettilineo uniforme ed il moto uniformemente accelerato. Ogni sezione comprende una parte introduttiva, nella quale vengono dati gli elementi fondamentali, ed una parte applicativa che permette la soluzione dei problemi particolari.

Nel moto rettilineo uniforme è messo in evidenza che spazi uguali vengono percorsi in tempi uguali e che lo spazio e il tempo sono direttamente proporzionali. La rappresentazione grafica di questa legge è una retta uscente dall'origine.

Nel caso di un moto uniformemente accelerato, con la tabulazione di alcuni valori discreti, si mette in evidenza che la velocità è direttamente proporzionale al tempo mentre lo spazio è direttamente proporzionale ai quadrati dei tempi.

Per la parte applicativa si usano le ben note formule riportate in ogni manuale di fisica, esse vengono illustrate durante l'esecuzione del programma. Merita un breve commento il modo di gestire l'input dei dati al computer.

Si è voluto poter calcolare una delle seguenti variabili: fissato un sistema di riferimento, x_0 è la posizione iniziale, x la posizione dopo il tempo t , v_0 è la velocità iniziale, v è la velocità al tempo t , a è l'accelerazione (costante), t il tempo. Ognuna di queste variabili si può calcolare in diversi modi a seconda dei dati a disposizione; ad esempio, si può calcolare x conoscendo x_0 , v , a , t oppure x_0 , v_0 , v ed a oppure x_0 , v_0 , t e v oppure, ancora, x_0 , v_0 , t ed a . Poiché ognuna delle sei variabili si può calcolare a partire da quattro diversi gruppi di dati iniziali, è ragionevole supporre di dover suddividere la sezione del programma in 24 parti per l'input dei dati e per i calcoli relativi. Ad un esame più attento si nota come sia possibile strutturare nello stesso modo l'input relativo a più variabili e creare delle routine.

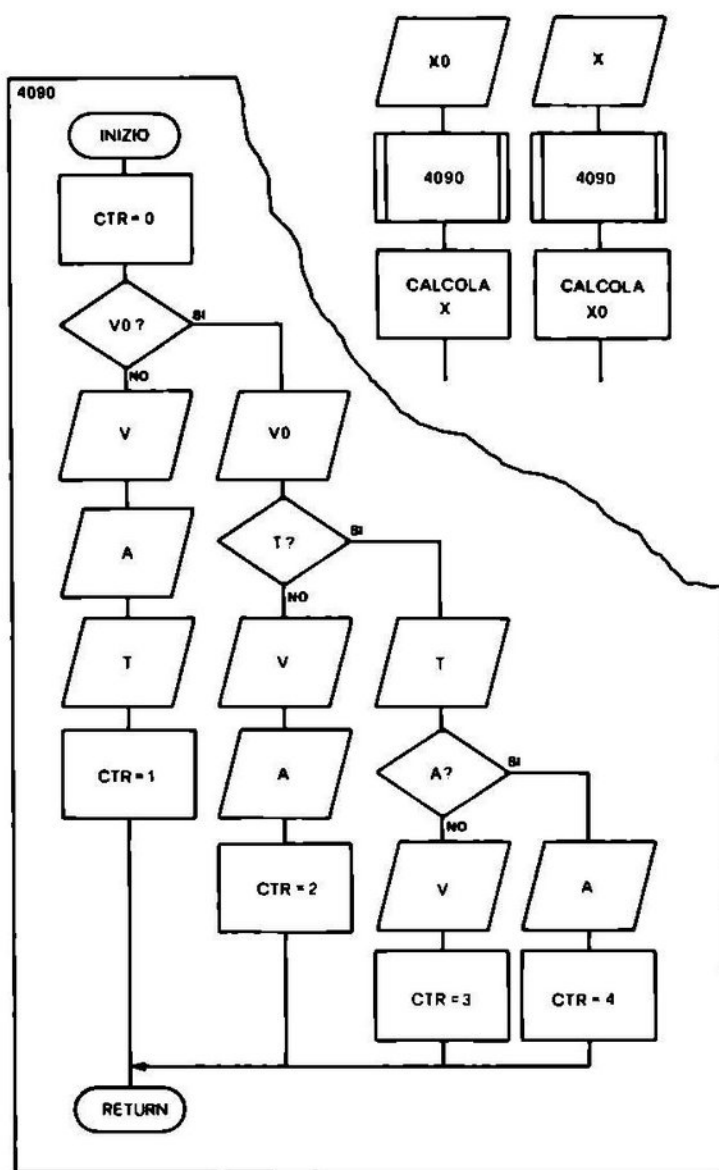


Fig. 23

La figura 23 mostra i diagrammi a blocchi per la richiesta dei dati necessari per il calcolo di x e x_0 , essi differiscono solo per l'input iniziale e tutta la parte restante può costituire una routine. La variabile contatore CTR, dopo il RETURN indica il tipo di calcolo da eseguire.

```

100 REM      CINEMATICA
110 REM      *****
120 REM      di Sergio Borsani
130 REM      versione: TI BASIC
140 CALL CLEAR
150 PRINT TAB(9);"CINEMATICA":;:
160 CALL CHAR(128,"0000000808087F08")
170 CALL CHAR(129,"0808083808080808")
180 CALL CHAR(130,"7808080808380808")
190 CALL CHAR(131,"0808780808080838")
200 CALL CHAR(132,"0808080878080808")
210 CALL CHAR(133,"0838080808087808")
220 CALL CHAR(134,"000000080808FF")
230 CALL CHAR(135,"00000001B3C3C18")
240 CALL CHAR(137,"0000183C3C18")
250 CALL CHAR(138,"00183C3C18")
260 CALL CHAR(139,"183C3C18")
270 CALL CHAR(140,"3C3C18")
280 CALL CHAR(141,"0000000000000018")
290 CALL CHAR(147,"000000FF")
300 CALL CHAR(150,"FF")
310 CALL CHAR(151,"08083E3E1C1C0808")
320 CALL CHAR(152,"0000000303CFF3C")
330 CALL CHAR(153,"3")
340 CALL CLEAR
350 PRINT "PREMI":;
360 PRINT "1. PER IL MOTO UNIFORME":;
370 PRINT "2. PER IL MOTO UNIFORMEMENTE"
380 PRINT "ACCELERATO":;
390 PRINT "3. PER LA CONVERSIONE DELLE"
400 PRINT "MISURE":;
410 PRINT "4. PER FINIRE":;
420 CALL KEY(0,K,B)
430 IF B=0 THEN 420
440 SC=K-48
450 IF (SC<1)+(SC>4) THEN 420
460 ON SC GOTO 470,1620,3530,5020
470 CALL CLEAR
480 PRINT "PREMI":;
490 PRINT "1. PER LE SPIEGAZIONI":;
500 PRINT "2. PER LE APPLICAZIONI":;
510 CALL KEY(0,K,B)
520 IF B=0 THEN 510
530 IF K=49 THEN 560
540 IF K=50 THEN 1120
550 GOTO 510
560 CALL CLEAR
570 PRINT "MOTO RETTILINEO UNIFORME":;:
580 CALL HCHAR(20,3,134,28)
590 CALL HCHAR(21,3,137)
600 GOSUB 3950
610 CALL HCHAR(23,3,32,28)
620 FOR I=1 TO 27
630 CALL HCHAR(20,2+I,32)
640 CALL HCHAR(20,3+I,137)
650 CALL SOUND(150,-4,1)
660 NEXT I
670 PRINT "SPAZI UGUALI IN TEMPI UGUALI":;
680 PRINT "VELOCITA' COSTANTE":;
690 GOSUB 3950
700 PRINT ""
710 PRINT "SCRIVI UN VALORE PER LA"
720 PRINT "VELOCITA', IN METRI AL SEC.":;
730 INPUT "V = M/S "V
740 PRINT ""
750 CALL HCHAR(23,3,147,28)
760 PRINT TAB(3);"TEMPO":TAB(11);"SPAZIO"
770 PRINT "(SECONDI)":TAB(10);"(METRI)":;
780 CALL HCHAR(24,3,147,28)
790 PRINT ""
800 FOR I=0 TO 10
810 PRINT TAB(4);I;TAB(12);I*V;TAB(18);"S/T ="I*V
820 NEXT I
830 PRINT ""
840 CALL HCHAR(24,3,147,28)
850 PRINT ""
860 GOSUB 3950

```

```

870 PRINT "1:"
880 CALL MCHAR(23,5,128)
890 CALL MCHAR(23,6,134,16)
900 CALL MCHAR(23,22,152)
910 PRINT TAB(3);CHR$(132);CHR$(138);TAB(14);"SPAZIO";CHR$(153)
920 FOR I=1 TO 12
930 PRINT TAB(3);CHR$(132);TAB(4+I);CHR$(138)
940 CALL SOUND(150,-4,1)
950 NEXT I
960 PRINT TAB(3);CHR$(151);"TEMPO":;
970 PRINT
980 PRINT "SPAZIO E TEMPO SONO:";"DIRETTAMENTE PROPORZIONALI":;
990 GOSUB 3930
1000 PRINT "1:"
1010 CALL CLEAR
1020 PRINT "VELOCITA'":;"V=S/T":;
1030 PRINT "SPAZIO":;"S=V*T":;
1040 PRINT "TEMPO":;"T=S/V":;
1050 PRINT "PIU' IN GENERALE, SE IL"
1060 PRINT "CORPO NON PARTE DALL'ORIGINE":;
1070 PRINT "X=X0+V*T":;
1080 PRINT "DOVE X0 E' LA POSIZIONE"
1090 PRINT "INIZIALE":;
1100 GOSUB 3930
1110 GOTO 340
1120 CALL CLEAR
1130 PRINT TAB(11);"X=X0+V*T":;
1140 PRINT "QUALE VARIABILE VUOI TROVARE"
1150 PRINT "X, X0, V OPPURE T?":;
1160 INPUT SCELTA$
1170 PRINT "1:"
1180 PRINT TAB(8);"SISTEMA MKS":;
1190 IF SCELTA$="X" THEN 1230
1200 IF SCELTA$="X0" THEN 1330
1210 IF SCELTA$="V" THEN 1440
1220 IF SCELTA$="T" THEN 1530
1230 PRINT "1:";"VARIABILE NON PREVISTA":;"RIPROVA":;
1240 GOTO 1130
1250 PRINT "1:"
1260 INPUT "X0 = ":X0
1270 INPUT "V = ":V
1280 INPUT "T = ":T
1290 PRINT
1300 X=X0+V*T
1310 PRINT "X = ";INT(1000*X)/1000;"METRI":;
1320 GOSUB 3990
1330 IF RISPOSTA$="S" THEN 1250 ELSE 340
1340 GOTO 340
1350 PRINT "1:"
1360 INPUT "X = ":X
1370 INPUT "V = ":V
1380 INPUT "T = ":T
1390 PRINT
1400 X0=X-V*T
1410 PRINT "X0 = ";INT(1000*X0)/1000;"METRI":;
1420 GOSUB 3990
1430 IF RISPOSTA$="S" THEN 1330 ELSE 340
1440 PRINT "1:"
1450 INPUT "X = ":X
1460 INPUT "X0 = ":X0
1470 INPUT "T = ":T
1480 PRINT
1490 V=(X-X0)/T
1500 PRINT "V = ";INT(1000*V)/1000;"M/S":;
1510 GOSUB 3990
1520 IF RISPOSTA$="S" THEN 1440 ELSE 340
1530 PRINT "1:"
1540 INPUT "X = ":X
1550 INPUT "X0 = ":X0
1560 INPUT "V = ":V
1570 PRINT
1580 T=(X-X0)/V
1590 PRINT "T = ";INT(1000*T)/1000;"SEC.":;
1600 GOSUB 3990
1610 IF RISPOSTA$="S" THEN 1530 ELSE 340
1620 CALL CLEAR
1630 PRINT "1:";"MOTO UNIFORMEMENTE"
1640 PRINT TAB(10);"ACCELERATO":;
1650 PRINT "1. SPIEGAZIONI":;
1660 PRINT "2. APPLICAZIONI":;
1670 INPUT "SCELTA?":SC
1680 IF SC=1 THEN 1720
1690 IF SC=2 THEN 2360
1700 PRINT "DEVI PREMERE 1 O 2, RIPETI":;
1710 GOTO 1670
1720 CALL CLEAR

```

```

1730 CALL HCHAR(5,10,130)
1740 FOR RIGA=6 TO 16 STEP 5
1750 FOR I=0 TO 2
1760 CALL HCHAR(RIGA+1,10,131+I)
1770 NEXT I
1780 CALL HCHAR(RIGA+3,10,129)
1790 CALL HCHAR(RIGA+4,10,130)
1800 NEXT RIGA
1810 CALL HCHAR(21,5,150,24)
1820 CALL HCHAR(4,12,135)
1830 GOSUB 3950
1840 CALL HCHAR(23,1,32,32)
1850 DATA 4,138,6,139,9,138,13,137,18,141,19,140
1860 RESTORE 1850
1870 FOR I=1 TO 6
1880 READ RIGA,CARAT
1890 CALL HCHAR(RIGA,12,CARAT)
1900 IF CARAT=141 THEN 1920
1910 CALL SOUND(300,-4,1)
1920 NEXT I
1930 INPUT "VUOI RIVEDERE? (S/N) ":RS
1940 IF RS="S" THEN 1720
1950 CALL CLEAR
1960 PRINT "TEMPO SPAZIO"
1970 PRINT "(SEC) (METRI)"
1980 PRINT
1990 CALL HCHAR(23,3,147,28)
2000 FOR I=1 TO 10
2010 PRINT I;TAB(9);4.9*I*I;TAB(18);" = 4.9*";STR$(I*I)
2020 NEXT I
2030 PRINT
2040 CALL HCHAR(23,3,147,28)
2050 PRINT
2060 PRINT "LO SPAZIO PERCORSO E'"
2070 PRINT "PROPORZIONALE AI QUADRATI"
2080 PRINT "DEI TEMPI"
2090 PRINT "LA LEGGE DEL MOTO E'::"
2100 PRINT TAB(8);"X=1/2*G*T^2 G=9.8":::
2110 GOSUB 3950
2120 CALL CLEAR
2130 PRINT "TEMPO VELOCITA'"
2140 PRINT "(SEC) (M/S)"
2150 CALL HCHAR(23,3,147,28)
2160 FOR I=1 TO 10
2170 PRINT I;TAB(12);9.8*I;TAB(18);" = 9.8*";STR$(I)
2180 NEXT I
2190 PRINT
2200 CALL HCHAR(23,3,147,28)
2210 PRINT
2220 PRINT "LA VELOCITA' E' PROPORZIONALE"
2230 PRINT "LE AL TEMPO TRASCORSO"
2240 PRINT "LA LEGGE DEL MOTO E'::"
2250 PRINT TAB(10);"V=G*T G=9.8":::
2260 GOSUB 3950
2270 CALL CLEAR
2280 PRINT "FORMULE DEL MOTO"
2290 PRINT "UNIFORMEMENTE ACCELERATO"
2300 PRINT "V=VO+A*T"
2310 PRINT "X=XO+VO*T+1/2*A*T^2"
2320 PRINT "V^2=VO^2+2*A*(X-XO)"
2330 PRINT "::::"
2340 GOSUB 3950
2350 IF RISP="S" THEN 1720 ELSE 340
2360 CALL CLEAR
2370 PRINT "VARIABILI"
2380 PRINT "X XO V VO A T"
2390 INPUT "COSA DEVI TROVARE? ":AS
2400 IF AS="X" THEN 2480
2410 IF AS="XO" THEN 2430
2420 IF AS="V" THEN 2470
2430 IF AS="VO" THEN 2490
2440 IF AS="A" THEN 3130
2450 IF AS="T" THEN 3260
2460 PRINT "ERRORE! RIPROVA"
2470 GOTO 2390
2480 PRINT "X="
2490 INPUT "XO = "
2500 GOSUB 4090
2510 ON CTR GOTO 2520,2540,2560,2580
2520 VO=V-A*T
2530 GOTO 2580
2540 T=(V-VO)/A
2550 GOTO 2580
2560 A=(V-VO)/T
2570 GOTO 2580
2580 X=XO+VO*T+A*T*T/2
2590 PRINT

```

```

2600 PRINT "X =" ; X ; "METRI" : : :
2610 GOSUB 3990
2620 IF RISP#="S" THEN 2360 ELSE 340
2630 PRINT : : :
2640 INPUT "X = " : X
2650 GOSUB 4090
2660 ON CTR GOTO 2670,2690,2710,2720
2670 V0=V-A*T
2680 GOTO 2720
2690 T=(V-V0)/A
2700 GOTO 2720
2710 A=(V-V0)/T
2720 X0=X-V0-T-A*T/2
2730 PRINT
2740 PRINT "X0 =" ; X0 ; "METRI" : : :
2750 GOSUB 3990
2760 IF RISP#="S" THEN 2360 ELSE 340
2770 PRINT : : :
2780 PRINT "CONOSCI T? (S/N)"
2790 CALL KEY (O,K,S)
2800 IF S=0 THEN 2790
2810 IF K=83 THEN 2880
2820 IF K=78 THEN 2840
2830 GOTO 2790
2840 GOSUB 4450
2850 V=V0+V0+2*A*X-2*A*X0
2860 V=SQR(ABS(V))
2870 GOTO 2950
2880 GOSUB 4500
2890 ON CTR GOTO 2900,2920,2940
2900 A=2*(X-X0-V0*T)/(T*T)
2910 GOTO 2940
2920 V0=(X-X0-A*T*T/2)/T
2930 GOTO 2940
2940 V=V0+A*T
2950 PRINT
2960 PRINT "V =" ; V ; "M/S" : : :
2970 GOSUB 3990
2980 IF RISP#="S" THEN 2360 ELSE 340
2990 PRINT : : :
3000 INPUT "A = " : A
3010 GOSUB 4760
3020 ON CTR GOTO 3030,3060,3080
3030 V0=V-2*A*X+2*A*X0
3040 V0=SQR(ABS(V0))
3050 GOTO 3090
3060 V0=(X-X0-A*T*T/2)/T
3070 GOTO 3090
3080 V0=V-A*T
3090 PRINT
3100 PRINT "V0 =" ; V0 ; "M/S" : : :
3110 GOSUB 3990
3120 IF RISP#="S" THEN 2360 ELSE 340
3130 PRINT : : :
3140 INPUT "V0 = " : V0
3150 GOSUB 4760
3160 ON CTR GOTO 3170,3190,3210
3170 A=(V-V0+V0)/(2*(X-X0))
3180 GOTO 3220
3190 A=2*(X-X0-V0*T)/(T*T)
3200 GOTO 3220
3210 A=(V-V0)/T
3220 PRINT
3230 PRINT "A =" ; A ; "M/S^2" : : :
3240 GOSUB 3990
3250 IF RISP#="S" THEN 2360 ELSE 340
3260 PRINT : : :
3270 PRINT "CONOSCI V? (S/N)"
3280 CALL KEY (O,K,S)
3290 IF S=0 THEN 3280
3300 IF K=83 THEN 3380
3310 IF K=78 THEN 3330
3320 GOTO 3280
3330 GOSUB 4450
3340 V=V0+V0+2*A*X-2*A*X0
3350 V=SQR(ABS(V))
3360 IF ((V-V0)/A)>=0 THEN 3380
3370 V=-V
3380 GOTO 3480
3390 INPUT "V = " : V
3400 GOSUB 4500
3410 ON CTR GOTO 3420,3440,3480
3420 A=(V-V-V0+V0)/(2*(X-X0))
3430 GOTO 3480
3440 V0=V-2*A*X+2*A*X0
3450 V0=SQR(ABS(V0))
3460 IF ((V-V0)/A)>=0 THEN 3480

```



```

3470 V0=-V0
3480 T=(V-V0)/A
3490 PRINT
3500 PRINT "T =";T;"SECONDI":;
3510 GOSUB 3990
3520 IF RISP="S" THEN 2360 ELSE 340
3530 CALL CLEAR
3540 PRINT "1. SECOND I IN ORE, MIN., SEC.":;
3550 PRINT "2. ORE, MIN., SEC. IN SECOND I":;
3560 PRINT "3. K/H IN M/S":;
3570 PRINT "4. M/S IN K/H":;
3580 CALL KEY(0,K,S)
3590 IF S=0 THEN 3580
3600 K=K-48
3610 IF (K<1)+(K>4) THEN 3580
3620 ON K GOTO 3630,3730,3830,3890
3630 CALL CLEAR
3640 INPUT "SECOND I? ":SEC
3650 PRINT :;
3660 H=INT(SEC/3600)
3670 SEC=SEC-H*3600
3680 M=INT(SEC/60)
3690 SEC=SEC-M*60
3700 PRINT H;"ORE "M;"MIN. "SEC;"SEC.":;
3710 GOSUB 3990
3720 IF RISP="S" THEN 3530 ELSE 340
3730 CALL CLEAR
3740 INPUT "ORE ":H
3750 SEC=H*3600
3760 INPUT "MIN. "M
3770 SEC=SEC+M*60
3780 INPUT "SEC. "S
3790 SEC=SEC+S
3800 PRINT :;"BOND: "SEC;"SECOND I":;
3810 GOSUB 3990
3820 IF RISP="S" THEN 3530 ELSE 340
3830 CALL CLEAR
3840 INPUT "VELOCITA' (K/H) ":V
3850 PRINT :;
3860 PRINT "E' UGUALE A "V/3.6;"M/S":;
3870 GOSUB 3990
3880 IF RISP="S" THEN 3530 ELSE 340
3890 CALL CLEAR
3900 INPUT "VELOCITA' (M/S) ":V
3910 PRINT :;
3920 PRINT "E' UGUALE A "V*3.6;"K/H":;
3930 GOSUB 3990
3940 IF RISP="S" THEN 3530 ELSE 340
3950 PRINT "PREMI UN TASTO"
3960 CALL KEY(0,K,S)
3970 IF S=0 THEN 3960
3980 RETURN
3990 PRINT "VUOI CONTINUARE? (S/N)"
4000 CALL KEY(0,K,S)
4010 IF S=0 THEN 4000
4020 IF (K=83)+(K=113) THEN 4050
4030 IF (K=78)+(K=110) THEN 4070
4040 GOTO 4000
4050 RISP="S"
4060 GOTO 4080
4070 RISP="N"
4080 RETURN
4090 CTR=0
4100 PRINT "CONOSCI V0? (S/N)"
4110 CALL KEY(0,K,S)
4120 IF S=0 THEN 4110
4130 IF K=83 THEN 4210
4140 IF K=78 THEN 4160
4150 GOTO 4110
4160 INPUT "V = ":V
4170 INPUT "A = ":A
4180 INPUT "T = ":T
4190 CTR=1
4200 GOTO 4440
4210 INPUT "V0 = ":V0
4220 PRINT "CONOSCI T? (S/N)"
4230 CALL KEY(0,K,S)
4240 IF S=0 THEN 4230
4250 IF K=83 THEN 4320
4260 IF K=78 THEN 4280
4270 GOTO 4230
4280 INPUT "V = ":V
4290 INPUT "A = ":A
4300 CTR=2
4310 GOTO 4440
4320 INPUT "T = ":T

```

```

4330 PRINT "CONOSCI A? (S/N) "
4340 CALL KEY(0,K,S)
4350 IF S=0 THEN 4340
4360 IF K=83 THEN 4420
4370 IF K=78 THEN 4390
4380 GOTO 4340
4390 INPUT "V = ":V
4400 CTR=3
4410 GOTO 4440
4420 INPUT "A = ":A
4430 CTR=4
4440 RETURN
4450 INPUT "VO = ":VO
4460 INPUT "A = ":A
4470 INPUT "X = ":X
4480 INPUT "XO = ":XO
4490 RETURN
4500 CTR=0
4510 PRINT "CONOSCI A? (S/N) "
4520 CALL KEY(0,K,S)
4530 IF S=0 THEN 4520
4540 IF K=83 THEN 4620
4550 IF K=78 THEN 4570
4560 GOTO 4520
4570 INPUT "X = ":X
4580 INPUT "XO = ":XO
4590 INPUT "VO = ":VO
4600 CTR=1
4610 GOTO 4750
4620 INPUT "A = ":A
4630 PRINT "CONOSCI VO? (S/N) "
4640 CALL KEY(0,K,S)
4650 IF S=0 THEN 4640
4660 IF K=83 THEN 4730
4670 IF K=78 THEN 4690
4680 GOTO 4640
4690 INPUT "X = ":X
4700 INPUT "XO = ":XO
4710 CTR=2
4720 GOTO 4750
4730 INPUT "VO = ":VO
4740 CTR=3
4750 RETURN
4760 CTR=0
4770 PRINT "CONOSCI T? (S/N) "
4780 CALL KEY(0,K,S)
4790 IF S=0 THEN 4780
4800 IF K=83 THEN 4880
4810 IF K=78 THEN 4830
4820 GOTO 4780
4830 INPUT "V = ":V
4840 INPUT "X = ":X
4850 INPUT "XO = ":XO
4860 CTR=1
4870 GOTO 5010
4880 INPUT "T = ":T
4890 PRINT "CONOSCI V? (S/N) "
4900 CALL KEY(0,K,S)
4910 IF S=0 THEN 4900
4920 IF K=83 THEN 4990
4930 IF K=78 THEN 4950
4940 GOTO 4900
4950 INPUT "X = ":X
4960 INPUT "XO = ":XO
4970 CTR=2
4980 GOTO 5010
4990 INPUT "V = ":V
5000 CTR=3
5010 RETURN
5020 CALL CLEAR
5030 END

```

Un esempio pratico di utilizzazione del programma farà comprendere il tipo di problemi risolvibili con esso e darà l'occasione per offrire qualche utile avvertimento.

Si voglia risolvere il seguente problema.

Dalla sommità di un edificio alto 100 m viene sparato un proiettile con una componente verticale della velocità di 98 m/s. Si chiede: (a) quale sarà la massima altezza raggiunta dal proiettile; (b) dopo quanto tempo

verrà raggiunta la massima altezza; (c) dopo quanto tempo toccherà il suolo; (d) quale sarà la sua velocità nella componente verticale, al momento dell'impatto al suolo.

Per rispondere alla prima domanda, si dovrà calcolare la variabile x . Selezionata la x quale variabile da calcolare, compare la richiesta del valore x_0 , senza questo dato il problema non è risolvibile. Poniamo $x_0 = 100$, con ciò è inteso che l'asse di riferimento è orientato verso l'alto. Viene rivolta la domanda: CONOSCI V_0 ? (S/N), la risposta è SI. Nell'input seguente, $V_0 =$, scriviamo 98 e alla successiva domanda, CONOSCI T ? (S/N), rispondiamo NO. Vengono allora chiesti i valori di V ed A . Deve essere $V = 0$, poiché alla massima altezza la componente verticale della velocità è zero e l'accelerazione è $A = -9.8$; il suo segno è negativo perché il verso è opposto rispetto all'asse di riferimento.

Vediamo che cosa apparirà sullo schermo.

RUN

```

VARIABILI:
X  X0  V  V0  A  T

COSA DEVI TROVARE? X          <ENTER>
X0 = 100                      <ENTER>
CONOSCI V0? (S/N)             <S>
V0 = 98                        <ENTER>
CONOSCI T? (S/N)              <N>
V = 0                          <ENTER>
A = -9.8                      <ENTER>

X = 590 METRI

VUOI CONTINUARE? (S/N)         <S>

```

La risposta è: $X = 590$ metri, si intende misurati dal suolo. Per calcolare T di cui al punto (b) del problema vi sarà il seguente output:

```

VARIABILI:
X  X0  V  V0  A  T

COSA DEVI TROVARE? T          <ENTER>
CONOSCI V? (S/N)              <S>
V = 0                          <ENTER>
CONOSCI A? (S/N)              <S>
A = -9.8                      <ENTER>
CONOSCI V0? (S/N)             <S>
V0 = 98                        <ENTER>

T = 10 SECONDI

VUOI CONTINUARE? (S/N)         <S>

```

Il programma in questo caso utilizza la formula $v = v_0 + at$, da cui $v_0 = v - at$. Per rispondere alla domanda del punto (c), si seleziona nuovamente la variabile T, in questo caso, tuttavia, non si conosce il valore di V nel momento T, pertanto avremo il seguente output:

```

VARIABILI:
X X0 V V0 A T
COSA DEVI TROVARE? T          <ENTER>

CONOSCI V? (S/N)              <N>
V0 = 9.8                       <ENTER>
A = -9.8                       <ENTER>
X = 0                          <ENTER>
X0 = 100                       <ENTER>

T = 20.97306535 SECONDI
VUOI CONTINUARE? (S/N)         <N>

```

Il computer calcola V con la formula $v^2 = v_0 + 2a(x - x_0)$ e sostituisce il valore nella $v = v_0 + at$ esplicitata rispetto a t. Una piccola complicazione è costituita dal fatto che V è uguale alla radice quadrata del valore V^2 e la funzione SQR restituisce solo la radice positiva, mentre ne esiste anche una negativa, opposta alla precedente, e molto spesso è proprio quella che va presa in considerazione. Nel programma si è adottata la soluzione, anche se poco ortodossa, di assumere prima il valore positivo di V e, se questo comporta un valore di T negativo, di prendere il suo opposto -V.

Moto ad accelerazione costante

Nella "Giornata Quarta" dei *Discorsi* (1638), dedicata al moto dei proiettili, Galileo Galilei enuncia il principio della composizione dei moti. Sotto una forza costante il moto di un corpo può essere visto come la composizione di un moto rettilineo uniforme, lungo la direzione perpendicolare alla forza, e di un moto uniformemente accelerato, nella direzione parallela alla forza (vedi figura 24).

Nel caso della forza di gravità si può ritenere, entro certi limiti, che l'accelerazione sia costante; il suo valore è 9.8 m/s^2 . Detta v_0 la velocità iniziale ed A l'angolo formato con la direzione orizzontale, le componenti della velocità sono $v_{0x} = v_0 \cos A$ e $v_{0y} = v_0 \sin A$. In ogni istante successivo risulta $v_x = v_{0x}$ e $v_y = v_{0y} - gt$. La posizione è determinata dalle coordinate:

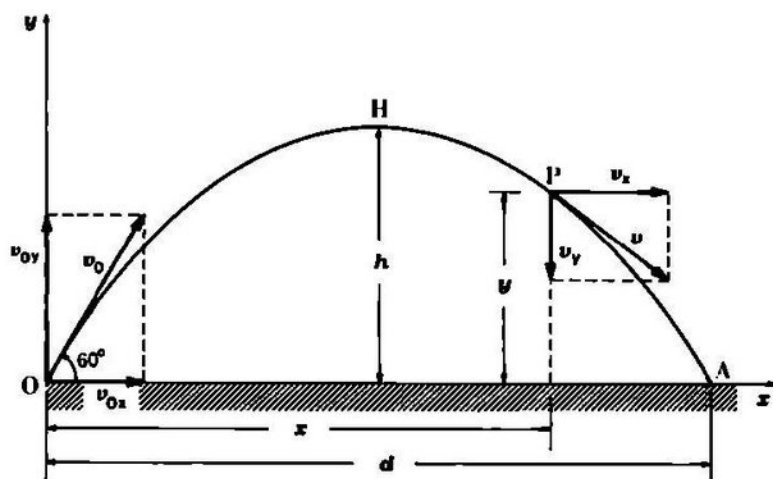


Fig. 24

$$x = v_{0x}t \quad \text{e} \quad y = y_0 + v_{0y}t - gt^2/2$$

Quest'ultima equazione, di secondo grado rispetto la variabile t , indica che la traiettoria è una parabola.

Il tempo di volo è $t = 2 v_{0y}/g$ e risulta il doppio del tempo impiegato per raggiungere la massima altezza $h = (v_0^2 \sin^2 A)/(2g)$. La gittata, cioè la massima distanza percorsa lungo l'asse orizzontale, può esser calcolata moltiplicando v_{0x} per il tempo di volo oppure con la formula

$$d = \frac{v_0^2 \sin 2A}{g}$$

Nel programma, se l'angolo A , formato dalla velocità iniziale con la direzione orizzontale, è diverso da zero, si assume che il corpo parta dall'origine, mentre se è $A=0$ viene chiesta l'altezza dalla quale parte il proiettile e si pone $y_0=h$. In quest'ultimo caso la traiettoria è un ramo di parabola e la posizione iniziale coincide con il vertice.

```

100 REM      BALISTICA
110 REM      *****
120 REM      di Sergio Borsani
130 REM      versione: TI BASIC
140 CALL CLEAR
150 PRINT TAB(11); "MOTO AD"
160 PRINT "ACCELERAZIONE COSTANTE"
170 DEF Y=Y0+V0Y*T-4.9*T*T
180 DIM S$(14,14),A$(20)
190 FOR J=0 TO 14
200 FOR K=0 TO 14
210 READ S$(J,K)

```



```

1090 PRINT TAB(17);CHR$(154)
1100 PRINT TAB(14);"VO";CHR$(151)
1110 PRINT TAB(15);CHR$(151)
1120 PRINT TAB(14);CHR$(151)
1130 PRINT TAB(13);CHR$(153);" A"
1140 PRINT TAB(12);CHR$(151);CHR$(152)
1150 CALL MCHAR(24,14,150,6)
1160 PRINT "!!!";
1170 PRINT "SCRIVI LA VELOCITA' INIZIALE";
1180 INPUT "VO = M/S ";VO
1190 PRINT "!!!";
1200 IF VO>0 THEN 1230
1210 PRINT "DEVE ESSERE VO>0, RIPETI";
1220 GOTO 1180
1230 PRINT "SCRIVI L'ANGOLO FORMATO CON"
1240 PRINT "L'ASSE ORIZZONTALE";
1250 INPUT "A = GRADI ";GRAD
1260 IF (GRAD>=0)*(GRAD<=90) THEN 1290
1270 PRINT "DEVE ESSERE 0<=A<=90, RIPROVA";
1280 GOTO 1250
1290 RAD=GRAD*3.141592654/180
1300 VOX=VO*COS(RAD)
1310 VOY=VO*SIN(RAD)
1320 PRINT "!!!";
1330 IF GRAD<>0 THEN 1420
1340 PRINT "SCRIVI L'ALTEZZA DA CUI VIE-"
1350 PRINT "NE EFFETTUATO IL LANCIO";
1360 INPUT "H = METRI ";H
1370 YO=H
1380 T=SQR(H/4.9)
1390 GIT=VOX*T
1400 PRINT "!!!";
1410 GOTO 1450
1420 H=VOX*VOX*SIN(RAD)^2/19.6
1430 GIT=VOX*VOX*SIN(2*RAD)/9.8
1440 YO=0
1450 PRINT "VUOI IL DIMENSIONAMENTO? AUTOMATICO? (S/N)";
1460 CALL KEY(0,K,B)
1470 IF S=0 THEN 1460
1480 IF K=83 THEN 1510
1490 IF K=78 THEN 1580
1500 GOTO 1460
1510 UX=GIT/29
1520 UY=H/19
1530 IF UX>UY THEN 1560
1540 U=UY
1550 GOTO 1720
1560 U=UX
1570 GOTO 1720
1580 CALL CLEAR
1590 CALL MCHAR(20,16,141,2)
1600 PRINT "ATTRIBUISCI UN VALORE AL"
1610 PRINT "SEGMENTO UNITARIO";
1620 INPUT "U = ";U
1630 PRINT "!!!";
1640 IF U>0 THEN 1670
1650 PRINT "DEVE ESSERE U>0, RIPETI";
1660 GOTO 1620
1670 PRINT "LO SCHERMO COPRE UN CAMPO DI"
1680 PRINT "U=19; PER U=29; METRI";
1690 INPUT "PROCEDO? (S/N) ";IR
1700 IF IR="S" THEN 1720
1710 GOTO 1580
1720 CALL CLEAR
1730 REM
1740 REM      ASSI DI
1750 REM      RIFERIMENTO
1760 REM
1770 CALL MCHAR(3,2,140)
1780 CALL MCHAR(3,3,141,28)
1790 CALL MCHAR(3,3,143)
1800 CALL VCHAR(4,2,142,18)
1810 CALL VCHAR(4,3,144,18)
1820 CALL MCHAR(22,2,145)
1830 CALL MCHAR(22,3,146,28)
1840 CALL MCHAR(22,3,147)
1850 SCALA$="U="&STR$(INT(1000*U)/1000)&" METRI"
1860 FOR J=1 TO LEN(SCALA$)
1870 L=ASC(SEG$(SCALA$,J,1))
1880 CALL MCHAR(2,1+J,L)
1890 NEXT J
1900 REM
1910 REM      STAMPA
1920 REM
1930 FOR COL=2 TO 31
1940 X=(COL-2)*U

```

```

1950 T=X/VOX
1960 PU=U/B
1970 PIXASS=173-INT(Y/PU)
1980 IF PIXASS<19 THEN 2190
1990 IF PIXASS>173 THEN 2190
2000 RIGA=INT((PIXASS-1)/8)+1
2010 PIXREL=PIXASS-((RIGA-1)*8)
2020 CARAT1=127+PIXREL
2030 CALL GCHAR(RIGA,COL,CARAT1)
2040 IF CARAT2=32 THEN 2070
2050 GOSUB 410
2060 CARAT1=NC
2070 CALL HCHAR(RIGA,COL,CARAT1)
2080 ON PIXREL GOTO 2090,2090,2190,2190,2190,2190,2190,2190,2190
2090 CARAT1=137+PIXREL
2100 RIGA=RIGA-1
2110 GOTO 2140
2120 CARAT1=129+PIXREL
2130 RIGA=RIGA+1
2140 CALL GCHAR(RIGA,COL,CARAT2)
2150 IF CARAT2=32 THEN 2180
2160 GOSUB 410
2170 CARAT1=NC
2180 CALL HCHAR(RIGA,COL,CARAT1)
2190 NEXT COL
2200 PRINT "PREMI 1. PER CONTINUARE"
2210 CALL KEY(0,K,S)
2220 IF S=0 THEN 2210
2230 IF K=49 THEN 2250
2240 GOTO 2210
2250 CALL CLEAR
2260 PRINT "PREMI:":
2270 PRINT "1. PER LA GITTATA, L'ALTEZZA"
2280 PRINT "2. PER LA MASSIMA ED IL TEMPO."
2290 PRINT "3. PER L'ALTEZZA IN DIPEN-"
2300 PRINT "4. DENZA DELLO SPAZIO (X).":
2310 PRINT "5. PER LA POSIZIONE E LA"
2320 PRINT "6. VELOCITA' IN FUNZIONE DEL"
2330 PRINT "TEMPO."
2340 PRINT "7. PER RIVEDERE IL GRAFICO."
2350 PRINT "8. PER RICOMINCIARE."
2360 PRINT "9. PER FINIRE."
2370 CALL KEY(0,K,S)
2380 IF S=0 THEN 2370
2390 K=K-8
2400 IF (K<1)+(K>6) THEN 2370
2410 ON K GOTO 2420,2490,2790,2940,1070,3070
2420 CALL CLEAR
2430 PRINT "GITTATA: ";BIT;"METRI":
2440 PRINT "ALTEZZA MASSIMA RAGGIUNTA: "
2450 PRINT "N° METRI":
2460 PRINT "TEMPO DI VOLO: ";BIT/VOX;"SECONDI":
2470 GOSUB 3020
2480 IF K=83 THEN 2250 ELSE 3070
2490 CALL CLEAR
2500 PRINT "IN QUESTA SEZIONE VIENE"
2510 PRINT "DATA L'ORDINATA DEL PUNTO"
2520 PRINT "MOBILE IN FUNZIONE DELLA"
2530 PRINT "ASCISSA (X).":
2540 INPUT "X = METRI " : X
2550 IF X=0 THEN 2590
2560 PRINT
2570 PRINT "DEVE ESSERE X>=0, RIPETI":
2580 GOTO 2540
2590 IF X<81 THEN 2640
2600 PRINT "X E' MAGGIORE DELLA GITTATA."
2610 PRINT "IL CORPO E' GIA' RICADUTO AL: "SUOLO":
2620 GOSUB 3020
2630 IF K=83 THEN 2540 ELSE 2250
2640 T=X/VOX
2650 PRINT "L'ORDINATA E':"
2660 PRINT "Y="METRI":
2670 PRINT "TEMPO TRASCORSO: ";T;"SECONDI":
2680 PRINT "COMPONENTI DELLA VELOCITA'":
2690 VV=VOX*8*T
2700 PRINT "VX ="VOX;"VY ="VY":
2710 V=SQR(VOX^2+VY^2)
2720 PRINT "VELOCITA' ":"INT(1000*V)/1000;"M/S":
2730 TA=VY/VOX
2740 PRINT "ANGOLO FORMATO DAL VETTORE"
2750 PRINT "VELOCITA' CON L'ASSE X:"
2760 PRINT "INT(ATN(TA)*57.2957795);"GRADI":
2770 GOSUB 3020
2780 IF K=83 THEN 2490 ELSE 2250
2790 CALL CLEAR
2800 PRINT "CALCOLO DELLA POSIZIONE AL"
2810 PRINT "TEMPO T (SECONDI)":

```



```

2820 INPUT "T = ":T
2830 IF T>=0 THEN 2860
2840 PRINT "DEVE ESSERE T>=0, RIPROVA":
2850 GOTO 2820
2860 IF T<GIT/VOX THEN 2910
2870 PRINT "DOPO TI SECONDI IL CORPO E'"
2880 PRINT "GIÀ CADUTO AL SUOLO":
2890 GOSUB 3020
2900 IF K=B3 THEN 2790 ELSE 2250
2910 PRINT "X=";VOX*T,"Y=";Y:
2920 GOTO 2680
2930 CALL CLEAR
2940 NC=96
2950 PRINT "VUOI CAMBIARE IL DIMENSIONAMENTO DEL GRAFICO? (S/N)"
2960 CALL KEY(O,K,S)
2970 IF S=0 THEN 2970
2980 IF K=B3 THEN 1580
2990 IF K=78 THEN 1720
3000 GOTO 2970
3010 PRINT "VUOI CONTINUARE? (S/N)"
3020 CALL KEY(O,K,S)
3030 IF S=0 THEN 3030
3040 IF (K=B3)+(K=78)=0 THEN 3030
3050 RETURN
3060 CALL CLEAR
3070 END

```

Il programma è strutturato in modo lineare. All'inizio è posta l'unica routine per la somma dei caratteri resa necessaria perché i punti che formano la traiettoria non cancellino parzialmente gli assi di riferimento. Di essa si parla più dettagliatamente all'inizio del capitolo dedicato alla grafica; la sua presenza rende un po' più lenta l'esecuzione della parte grafica ma i risultati sono indubbiamente migliori e all'aspetto più completi.

Seguono l'input dei dati, costituiti dalla velocità iniziale, dall'angolo ed eventualmente dall'altezza dalla quale viene effettuato il lancio, il dimensionamento automatico, su richiesta, la stampa dei punti che costituiscono la traiettoria e, infine, i calcoli e la stampa dei risultati numerici.

Per la grafica sono stati definiti caratteri speciali per ottenere gli assi di riferimento ed i punti della traiettoria. I punti sono in realtà piccoli cerchi, la loro posizione è comunque determinata dal pixel centrale. La traiettoria viene visualizzata ponendo un cerchietto in ogni colonna, da colonna 2 a colonna 31; la loro ordinata varia tuttavia pixel per pixel, poiché sono stati definiti cerchietti a diverse altezze all'interno di un carattere ed alcuni occupano una posizione "a cavallo" tra una riga e la successiva.

Due ultimi brevi consigli all'utilizzatore: (1) angoli prossimi a 90° rendono inadeguata la rappresentazione grafica, si può allora dimensionare a piacimento il grafico cercando di ottenere almeno una parte della traiettoria; (2) il dimensionamento automatico comporta, generalmente, un valore decimale per il segmento unitario, rendendo meno immediata la determinazione delle misure sul grafico.

Poiché il valore unitario che si ottiene con il dimensionamento automatico viene mostrato, è possibile creare una seconda volta il grafico dimensionandolo in modo non automatico ed aumentando la misura del segmento unitario fino ad un valore intero desiderato.

Orbite dei satelliti artificiali

Nel 1680 Robert Hooke sottopose a Newton l'ipotesi che il moto di un satellite potesse essere il risultato della composizione di un moto inerziale, lungo la tangente, e di un moto accelerato, lungo la direzione radiale. Esistono analogie con il moto di un proiettile esaminato in precedenza; la differenza consiste nel fatto che mentre prima eravamo in presenza di una forza costante, nel nostro caso la forza è centrale e varia di intensità e direzione.

Da un punto di vista teorico, un'orbita planare è interamente determinata conoscendo l'altezza e la velocità tangenziale iniziali. La massa del satellite, trascurabile rispetto la terra, non ha influenza sulla traiettoria descritta. Le dimensioni dell'orbita (semiasse maggiore) dipendono dall'energia, mentre l'eccentricità dipende sia dall'energia che dalla quantità di moto.

L'energia totale del satellite è:

$$E = \frac{1}{2} m v_0^2 - \frac{\gamma m M}{R_T + h}$$

dove m è la massa del satellite, $M = 5.98 \times 10^{24}$ kg è la massa della terra, v_0 è la velocità tangenziale nel momento iniziale, γ è la costante gravitazionale, 6.67×10^{-11} nel sistema MKS, $R_T = 6.37 \times 10^6$ è il raggio terrestre, h è la quota iniziale.

Il momento della quantità di moto è:

$$L = m(R_T + h)v_0$$

Entrambe le grandezze, E ed L , sono costanti, fissate le condizioni iniziali.

La prima legge di Keplero dice che l'orbita di un satellite è un'ellisse e che la terra occupa uno dei fuochi.

La seconda legge di Keplero afferma che le aree "spazzate" in tempi uguali dal raggio vettore che congiunge la terra al satellite sono uguali. Pertanto, immaginando il raggio vettore come una lancetta impennata al centro della terra, lo vedremo ruotare rapidamente quando il satellite si trova vicino alla terra (perigeo), mentre ruoterà lentamente quando sarà lontano (apogeo).

Una costante C , detta velocità areolare, rappresenta il doppio dell'area spazzata nell'unità di tempo.

In questo programma C è stata posta arbitrariamente uguale a $T \times 10^{11}$; variando T da tastiera si ottiene un moto più lento o più rapido, poiché così si fa variare l'area spazzata nell'unità di tempo. L'esperienza dette-

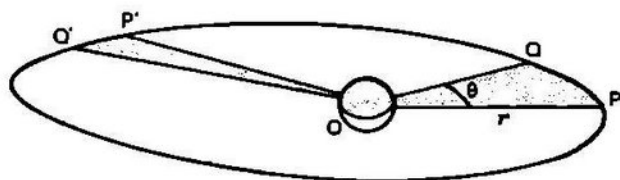


Fig. 25

rà di usare valori bassi di T ($T < 5$) per orbite di piccole dimensioni e valori alti ($T > 5$) per orbite ampie.

L'ellisse è una curva generata da un punto che si muove in modo tale che il rapporto fra le distanze da un punto fisso, detto fuoco, ed una retta, detta direttrice, sia costante e minore di 1. Questa costante è detta eccentricità.

Da un punto di vista geometrico (vedi figura 26) l'eccentricità è: $e = PF/PQ$.

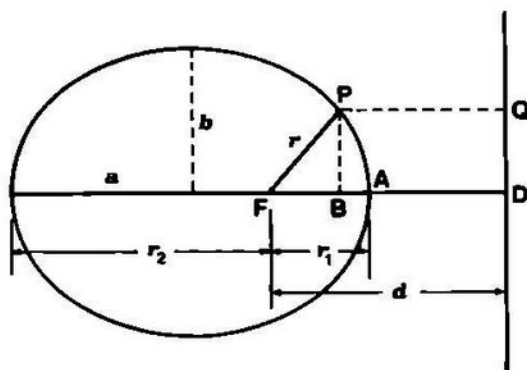


Fig. 26

In riferimento al moto di un satellite l'eccentricità è data dalla formula:

$$e = \sqrt{1 + \frac{2E}{m} \left(\frac{L}{GmM} \right)^2}$$

dove E è l'energia totale ed L il momento della quantità di moto.

Calcolata l'eccentricità dell'orbita e noto il valore di r_1 ($r_1 = R_T + h$), si deduce la misura di FD , cioè la distanza della direttrice dal fuoco.

$$d = r_1 + \frac{r_1}{e}$$

La distanza angolare percorsa nell'unità di tempo varia da punto a punto. Chiamato RAD l'angolo in radianti, sarà $RAD = C/R^2$ dove R rappresenta la distanza del satellite dal centro della terra. Tale distanza è inizialmente nota, per i punti successivi vale la seguente formula:

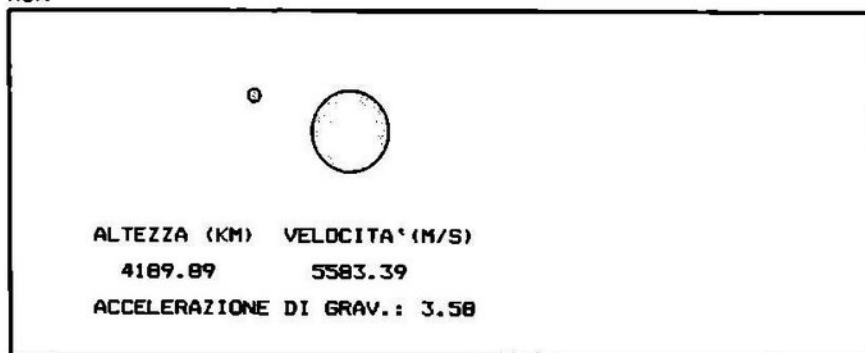
$$R = \frac{e \times d}{1 + e \times \cos(RAD)}$$

Questo calcolo è effettuato alla linea 1390 del programma. L'ultimo valore di R darà il nuovo valore di RAD e questo il successivo valore di R , e così via. Le coppie di valori di R e RAD costituiscono le coordinate polari dei punti che formano l'orbita. Le coordinate cartesiane si trovano facilmente: $PB = R \times \sin(RAD)$, $FB = R \times \cos(RAD)$. Successivamente è necessario trasformare le misure del sistema MKSA a quello relativo ai pixel del video e traslare l'origine per adeguarsi alle coordinate usate dal sistema operativo del computer.

Si è adottato che un pixel corrisponda a 500 km, allora, detta PIXRIGA il numero di riga in pixel, è $PIXRIGA = 97 - PB \cdot 2 \cdot 10^{-6}$ e, detto PIXCOLO il numero di colonna in pixel, è $PIXCOLO = 161 + FB \cdot 2 \cdot 10^{-6}$.

Durante l'esecuzione simulata del volo orbitale l'utente ha a disposizione alcuni tasti programmati. Nell'ordine, il tasto 1 provoca l'apparizione della traccia dell'orbita sotto forma di puntini bianchi; il tasto 2 fa scomparire la traccia dell'orbita pur restando visibile il satellite ed inalterato il suo movimento; il tasto 9 fa tornare al menù principale; il tasto 0 prepara il computer per l'introduzione di dati relativi ad una nuova orbita; infine, qualsiasi altro tasto provoca l'arresto del movimento e l'apparizione dei dati riguardanti l'altezza, la velocità ed il valore dell'accelerazione di gravità riferiti a quell'istante.

RUN



I punti che tracciano l'orbita sono ottenuti con degli sprite di colore bianco. Con il TI Extended Basic è consentito l'uso di un massimo di 28 sprite e non ne possono coesistere più di quattro sulla stessa riga. Per questi motivi, a volte, qualche punto può non apparire oppure può scomparire e riapparire al passaggio del satellite. Regolando il tempo relativo di scansione T è possibile non solo controllare la velocità del satellite ma anche una più appropriata distribuzione dei punti che tracciano l'orbita. In ogni caso i punti saranno più densi in apogeo e più distanziati in perigeo, in accordo con la seconda legge di Keplero.

Nell'ultima parte del programma vengono elaborati i dati relativi ad orbite circolari. Scritta la quota in input, in computer fornisce la velocità orbitale e la velocità di fuga, sia in metri al secondo che in chilometri all'ora, ed il periodo di rivoluzione in ore, minuti e secondi.

È facile verificare che ad una quota di 36000 km il periodo di rivoluzione è di circa un giorno, confermando che l'orbita è geostazionaria.

L'esempio che segue si riferisce ad un'orbita circolare posta a 500 km di altezza dal suolo.

RUN

```

- ORBITE CIRCOLARI -

ALTEZZA = KM 500                <ENTER>

VELOCITA' ORBITALE
M/S 7619.030223
KM/H 27428.5068

VELOCITA' DI FUGA
M/S 10774.93587
KM/H 38789.76915

PERIODO ORBITALE
GIORNI ORE MINUTI SECONDI
0      1      34      25

1. PER CONTINUARE
2. PER FINIRE

```

```

100 REM *****
110 REM *
120 REM *      ORBITE      *
130 REM *    DEI SATELLITI *
140 REM *    ARTIFICIALI  *
150 REM *
160 REM *****
170 REM di SERGIO BORSANI
180 REM versione: EXT.BASIC
190 CALL CLEAR
200 CALL DELSPRITE(ALL)
210 REM ***DEFINIZIONE CARATTERI***
220 CALL CHAR(128,"00030F1F3F3F7F7F")
230 CALL CHAR(129,"00E0FBFCFEFEFF")
240 CALL CHAR(130,"FFFFFFEFCFCF8E0")
250 CALL CHAR(131,"7F7F7F3F3F1F0F03")
260 CALL CHAR(136,"8")
270 CALL COLOR(13,3,1)

```

```

280 DISPLAY AT(11,9):"ORBITA DI"
290 DISPLAY AT(13,8):"UN SATELLITE"
300 CALL COLOR(2,4,16)
310 CALL HCHAR(9,8,42,16)
320 CALL VCHAR(10,8,42,5)
330 CALL VCHAR(10,23,42,5)
340 CALL HCHAR(15,8,42,16)
350 FOR J=1 TO 500 : NEXT J
360 REM *** MENU ***
370 PRINT "1: STUDIO DI UN'ORBITA:" " "
380 PRINT "2: VELOCITA' ORBITALI" " " E DI FUGA ":" "
390 PRINT "3: FINE:" " "
400 DISPLAY AT(24,1):"SCELTA:"
410 ACCEPT AT(24,9):BEEP VALIDATE("123"):SC$
420 CALL CLEAR
430 CALL COLOR(2,2,1)
440 IF SC$=CHR$(50) THEN 1800
450 IF SC$=CHR$(51) THEN 2290
460 CALL DELSPRITE(ALL)
470 REM *** INPUT DATI ***
480 PRINT "STUDIO DI UN'ORBITA:" " ":" ":" "
490 PRINT "MASSA DEL SATELLITE"
500 INPUT "M = KG ":"KG
510 PRINT " "
520 PRINT "ALTEZZA INIZIALE"
530 INPUT "H = KM ":"H
540 IF H>0 THEN 580
550 PRINT " " "L'ALTEZZA NON PUO' ESSERE"
560 PRINT "NEGATIVA! RIPROVA:" " "
570 GOTO 530
580 IF H<47500 THEN 610
590 PRINT " " "SPIACENTE LA QUOTA E' TROPPO ALTA PER QUESTO PROGRAMMA!"
600 PRINT " " "1. GOTO 520
610 PRINT " " "
620 PRINT "VELOCITA' TANGENZIALE"
630 INPUT "V = M/S ":"V
640 PRINT " " " "TEMPO RELATIVO (DA 1 A 9)"
650 INPUT "T = ":"T
660 IF T>=1 OR T<=10 THEN 680
670 PRINT "VALORE NON AMMESSO, RIPETI" " " GOTO 650
680 CALL CLEAR
690 REM *** CALCOLI ***
700 R1=6.37E+6+H*1000
710 L=KG*V*H
720 BM12=3.988E+14*KG
730 ENERGIA=(KG/2)*V^2-GM12/R1
740 E=8DR(1+2*ENERGIA/KB*(L/BM12)^2)
750 VC=SGR(3.988E+14/R1)
760 IF V<VC THEN 840
770 D=R1+R1/E
780 R2=E*D/(1-E)
790 A=(R1+R2)/2
800 IF E>=1 THEN 910
810 B=A*SGR(1-E^2)
820 PERIPX=R1*2*1E-6
830 GOTO 910
840 D=R1/E-R1
850 R2=E*D/(1+E)
860 A=(R1+R2)/2
870 IF E>=1 THEN 910
880 B=A*SGR(1-E^2)
890 PERIPX=R2*2*1E-6
900 REM *** DISPLAY ***
910 PRINT "SISTEMA MKSA:" " "
920 PRINT "V (INIZ.) =" "V0:" "
930 PRINT "V (CIRC.) =" "VC:" "
940 PRINT "ENERGIA TOT. =" "ENERGIA:" "
950 PRINT "MOMENTO DELLA QUANTITA' DI MOTO =" "L:" "
960 IF E>=1 THEN 990
970 PRINT "SEMIASSE MAG. =" "A:" "
980 PRINT "SEMIASSE MIN. =" "B:" "
990 PRINT "ECCENTRICITA' =" "E:" "
1000 IF V<VC THEN X1=R2 : X2=R1 : GOTO 1020
1010 X1=R1 : X2=R2
1020 PRINT USING "H IN PERIGEO = #####.##":X1-6.37E+6
1030 PRINT
1040 IF E<1 THEN 1070
1050 PRINT "L'ORBITA E' APERTA!" " "
1060 GOTO 1120
1070 PRINT USING "H IN APOGEO = #####.##":X2-6.37E+6
1080 PRINT
1090 IF R1<6.3E+6 OR R2<6.3E+6 THEN 1110
1100 GOTO 1120
1110 PRINT "VELOCITA' TROPPO BASSA:"
1120 PRINT " " * PREMI UN TASTO
1130 CALL KEY(0,K,B)
1140 IF B=0 THEN 1130

```

```

1150 CALL CLEAR
1160 IF E>=1 THEN 520
1170 REM *** TERRA ***
1180 CALL HCHAR(12,20,128)
1190 CALL HCHAR(12,21,129)
1200 CALL HCHAR(13,21,130)
1210 CALL HCHAR(13,20,131)
1220 REM *** SATELLITE ***
1230 CALL Ssprite(01,136,2,97,161+PERIPX)
1240 NS=1
1250 C=T*1E+13
1260 RAD=C/R1^2
1270 IF V0>=VC THEN 1290
1280 RAD=RAD+PI
1290 IF RAD<1.5E+10 THEN 1350
1300 PRINT "SPIACENTE, CON QUESTI VALORI"
1310 PRINT "NON POSSO DISEGNARE L'ORBITA"
1320 CALL CLEAR
1330 GOTO 440
1340 REM *** DETERMINAZIONE POSIZIONE SATELLITE ***
1350 R=E*D/(1+E*COB(RAD))
1360 PB=R*SIN(RAD)
1370 FB=R*COB(RAD)
1380 PIXRIGA=97-PB*2*1E-6
1390 PIXCOLD=161+FB*2*1E-6
1400 IF PIXRIGA<1 OR PIXRIGA>192 THEN 1430
1410 IF PIXCOLD<1 OR PIXCOLD>236 THEN 1430
1420 CALL LOCATE(01,PIXRIGA,PIXCOLD)
1430 RAD=RAD+C/R1^2
1440 REM *** TASTI PROGRAMMATI ***
1450 CALL KEY(0,K,B)
1460 IF S=0 THEN 1520
1470 IF K=48 THEN BCS=CHR$(49):: GOTO 420
1480 IF K=49 THEN 1690
1490 IF K=50 THEN 1750
1500 IF K=57 THEN CALL CLEAR :: CALL DELSPRITE(ALL):: GOTO 370
1510 GOSUB 1540
1520 GOTO 1350
1530 REM *** DISPLAY DATI ***
1540 DISPLAY AT(20,1):"ALTEZZA(KM) VELOCITA'(M/S)"
1550 CALL POSITION(01,XX,YY)
1560 HX=R/1000-6370
1570 VX=L/(KB*R)
1580 ACC=3.988E+14/R^2
1590 DISPLAY AT(22,3):SIZE(10):USING "#####.###":HX
1600 DISPLAY AT(22,15):USING "#####.###":VX
1610 DISPLAY AT(24,1):"ACCELERAZIONE DI BRAV.1"
1620 DISPLAY AT(24,24):USING "#####.###":ACC
1630 CALL KEY(0,K,B)
1640 IF S=0 THEN 1630
1650 CALL HCHAR(20,32,160)
1660 CALL LOCATE(01,XX,YY)
1670 RETURN
1680 REM *** TRACCIA ORBITA ***
1690 NS=NS+1
1700 IF NS>28 THEN 1350
1710 CALL POSITION(01,XX,YY)
1720 CALL Ssprite(NS,136,16,XX,YY)
1730 GOTO 1350
1740 REM *** CANCELLA ORBITA ***
1750 CALL POSITION(01,XX,YY)
1760 CALL DELSPRITE(ALL)
1770 CALL Ssprite(01,136,2,XX,YY)
1780 NS=1
1790 GOTO 1350
1800 REM *** VELOCITA' ORBITALI ***
1810 REM *** E VELOCITA' DI FUGA ***
1820 PRINT TAB(6):"DATI COSTANTI": " " " " "
1830 PRINT "COSTANTE GRAVITAZIONALE"
1840 PRINT "G = 6.67E-11"
1850 PRINT "RAGGIO MEDIO TERRESTRE"
1860 PRINT "R = 6.37E+6"
1870 PRINT "MASSA DELLA TERRA"
1880 PRINT "M = 5.98E+24"
1890 PRINT " (SISTEMA MKSA)"
1900 PRINT "S PREMI UN TASTO"
1910 CALL KEY(0,K,B)
1920 IF S=0 THEN 1910
1930 CALL CLEAR
1940 DISPLAY AT(3,5):"-- ORBITE CIRCOLARI --"
1950 DISPLAY AT(6,1):"ALTEZZA = KM"
1960 DISPLAY AT(8,1):"VELOCITA' ORBITALE"
1970 DISPLAY AT(12,1):"VELOCITA' DI FUGA"
1980 DISPLAY AT(16,1):"PERIODO ORBITALE"
1990 DISPLAY AT(17,3):"GIORNI ORE MINUTI SECONDI"
2000 ACCEPT AT(6,14):VALIDATE(NUMERIC)BEEP:HH

```

```

2010 IF HH<0 THEN 2000
2020 R2R=SGR(HH*1000+6.37E+6)
2030 R2GM=1.997E+7
2040 V1=R2GM/R2R
2050 V2=V1*SGR(2)
2060 PER=(HH*1000+6.37E+6)*2*PI/V1
2070 BG=INT(PER/86400)
2080 RESTO=PER-BG*86400
2090 HD=INT(RESTO/3600)
2100 RESTO=RESTO-HD*3600
2110 MI=INT(RESTO/60)
2120 DISPLAY AT(9,3): "M/S " ; V1
2130 DISPLAY AT(10,3): "KM/H " ; V1*3.6
2140 DISPLAY AT(13,3): "M/S " ; V2
2150 DISPLAY AT(14,3): "KM/H " ; V2*3.6
2160 DISPLAY AT(18,3): BG ; " " ; HD ; " " ; MI ; " " ; INT(RESTO-MI*60)
2170 DISPLAY AT(23,1): "1. PER CONTINUARE"
2180 DISPLAY AT(24,1): "2. PER FINIRE"
2190 CALL KEY(0,K,S)
2200 IF S=0 THEN 2190
2210 IF K=49 THEN 2230
2220 IF K=50 THEN CALL CLEAR :: GOTO 370 ELSE 2190
2230 CALL MCHAR(9,1,32,32)
2240 CALL MCHAR(10,1,32,32)
2250 CALL MCHAR(13,1,32,32)
2260 CALL MCHAR(14,1,32,32)
2270 CALL MCHAR(18,1,32,224)
2280 GOTO 2000
2290 CALL CLEAR
2300 END

```

Il programma è strutturato in modo semplice e lineare. Nelle righe 100-500 si definiscono i caratteri che formano la terra e viene dato il menù per l'accesso alle due principali sezioni. Nella parte 510-720 si accettano i dati in input e successivamente, nelle linee 730-930, si effettuano i calcoli per le caratteristiche dell'orbita i quali vengono visualizzati nell'intervallo 940-1210. I dati comprendono la velocità iniziale, la velocità relativa ad un'orbita circolare, l'energia totale (il segno negativo indica che siamo in presenza di una forza attrattiva), il momento della quantità di moto, lunghezza del semiasse maggiore e del semiasse minore, eccentricità, altezza in perigeo ed in apogeo. Le linee 1220-1270 danno il grafico della terra e del satellite; la terra è fatta apparire più piccola di quanto risulta dalla scala delle grandezze poiché, altrimenti, le orbite più basse sembrerebbero intersecare la terra stessa.

In 1280-1370 si determina il valore di RAD (incremento angolare in radianti); tale valore non può essere maggiore di $1.570E+10$, limite massimo per l'argomento della successiva funzione $\cos(\text{RAD})$. Le linee 1380-1470 posizionano il satellite ed in 1480-1560 sono programmati i tasti per le funzioni esposte in precedenza. Nelle parti 1570-1710, 1720-1770 e 1780-1830 trovano posto rispettivamente tre routine per la scrittura dei dati temporanei, per tracciare l'orbita e per cancellarla.

Nell'ultima parte, linee 1840 e seguenti, vengono elaborati i dati riferiti ad orbite circolari.

La memoria richiesta è di 6.4 K.

Conversione delle unità di misura

In appendice al capitolo viene presentato questo semplice programma che permette di convertire le principali unità di misura adottate nel Sistema Internazionale, nel sistema cgs, in quello tecnico ed in quello in uso nei paesi anglosassoni. Le grandezze fisiche considerate sono: lunghezza, tempo, angoli, velocità, accelerazione, massa, forza, pressione, temperatura, energia, potenza. Le tabelle di conversione non vengono riportate nel testo per ragioni di spazio e perché sono facilmente desumibili dal listato in corrispondenza delle istruzioni DATA presenti nelle rispettive sezioni del programma. A seconda della grandezza fisica prescelta, i fattori di conversione vengono caricati nella matrice bidimensionale A(I,J) e richiamati in base all'unità di misura I nella quale è espressa la grandezza e all'unità J nella quale deve essere convertita.

Per il lettore che non possiede ancora grande padronanza del Basic si ricorda che l'istruzione RESTORE serve a riportare il puntatore sui dati che devono esser letti con la successiva READ, altrimenti questi vengono letti sequenzialmente, indipendentemente dalla loro posizione nel programma.

Poiché le varie sezioni posseggono la stessa struttura, si fa il massimo uso di routine, per l'input, il calcolo e la stampa dei risultati. Risulta inoltre agevole trascrivere o trasferire ad un altro programma anche solo la parte che interessa, senza particolari adattamenti.

```

100 REM CONVERSIONI DELLE
110 REM UNITA' DI MISURA
120 REM *****
130 REM di: SERGIO BORSANI
140 REM versione: TI BASIC
150 CALL CLEAR
160 CALL CHAR(128,"F")
170 CALL COLOR(13,6,6)
180 PRINT TAB(8);"CONVERSIONI DI";TAB(7);"UNITA' DI MISURA";:;:;:;:;:;:
190 CALL MCHAR(10,7,128,20)
200 CALL VCHAR(11,7,128,5)
210 CALL VCHAR(11,26,128,5)
220 CALL MCHAR(16,7,128,20)
230 CALL CHAR(136,"003048483")
240 FOR T=1 TO 500
250 NEXT T
260 CALL CLEAR
270 PRINT "SCEGLI IL TIPO DI GRANDEZZA"
280 PRINT "SCRIVENDO IL NUMERO CORRI-": "SPONDENTE.":
290 PRINT " 1. TEMPO": " 2. LUNGHEZZA": " 3. ANGOLI"
300 PRINT " 4. VELOCITA'": " 5. ACCELERAZIONE": " 6. MASSA"
310 PRINT " 7. FORZA": " 8. PRESSIONE": " 9. ENERGIA"
320 PRINT "10. POTENZA": "11. TEMPERATURA":
330 INPUT "SCELTA?":IN
340 IF (IN>11)&(IN<=11) THEN 370
350 PRINT TAB(6);"- ERRORE! RIPETI -":
360 GOTO 330
370 ON N GOTO 380,560,760,930,1110,1280,1450,1620,1800,1990,2160
380 CALL CLEAR
390 PRINT "TEMPO":
400 GOSUB 2690
410 DATA SECONDI,MINUTI,ORE,GIORNI,ANNI
420 RESTORE 410
430 MAX=5
440 GOSUB 2840
450 DATA 1,1.667E-2,2.778E-4,1.158E-5,3.169E-8
460 DATA 60,1.1.667E-2,6.945E-4,1.9E-6
470 DATA 3600,60,0.0417,2.14E-4
480 DATA 86400,1440,24,1,2.74E-3

```

```

490 DATA 3.154E+7,5.259E+5,8760,365,1
500 GOSUB 2880
510 GOSUB 2720
520 GOSUB 2600
530 GOSUB 2780
540 GOSUB 2470
550 IF K=78 THEN 270 ELSE 510
560 CALL CLEAR
570 PRINT "LUNGHEZZA":
580 GOSUB 2690
590 DATA METRO, YARD, FOOT, INCH, MIGLIO, MIGLIO MARINO, ANNO LUCE
600 RESTORE 590
610 MAX=7
620 GOSUB 2840
630 DATA 1,1.094,3.28,39.37,6.214E-4,5.4E-4,1.057E-16
640 DATA .914,1.3,36,5.67E-4,4.93E-4,9.66E-17
650 DATA .305,1.333,1.12,1.89E-4,1.64E-4,3.22E-17
660 DATA .0254,2.777E-2,8.333E-2,1.57E-5,1.37E-5,2.7E-18
670 DATA 1609.344,1760,5280,63360,1.8684,1.7E-13
680 DATA 1853.184,2026.666,6080,72960,1.1515,1.1957E-13
690 DATA 9.46E+15,1.035E+16,3.1E+16,3.7E+17,3.88E+12,5.1E+12,1
700 GOSUB 2880
710 GOSUB 2720
720 GOSUB 2600
730 GOSUB 2780
740 GOSUB 2470
750 IF K=83 THEN 710 ELSE 270
760 CALL CLEAR
770 PRINT "ANGOLI":
780 GOSUB 2690
790 DATA RADIANTI, GRADI, PRIMI, SECONDI
800 RESTORE 790
810 MAX=4
820 GOSUB 2840
830 DATA 1,57.3,3438,206280
840 DATA 1.743E-2,1.60,3600
850 DATA 2.908E-4,1.666E-2,1.60
860 DATA 4.847E-6,2.777E-4,1.666E-2,1
870 GOSUB 2880
880 GOSUB 2720
890 GOSUB 2600
900 GOSUB 2780
910 GOSUB 2470
920 IF K=83 THEN 880 ELSE 270
930 CALL CLEAR
940 PRINT "VELOCITA":
950 GOSUB 2690
960 DATA M/SEC, CM/SEC, KM/H, FT/SEC, NODI
970 RESTORE 960
980 MAX=5
990 GOSUB 2840
1000 DATA 1,100,3.6,3.281,1.944
1010 DATA 1E-2,1.3,6E-2,3.28E-2,1.944E-2
1020 DATA .2778,27.778,1.911,3397
1030 DATA .3048,30.48,1.094,1.3924
1040 DATA .5148,51.48,1.853,1.689,1
1050 GOSUB 2880
1060 GOSUB 2720
1070 GOSUB 2600
1080 GOSUB 2780
1090 GOSUB 2470
1100 IF K=83 THEN 1060 ELSE 270
1110 CALL CLEAR
1120 PRINT "ACCELERAZIONE":
1130 GOSUB 2690
1140 MAX=4
1150 DATA M/SEC AL SEC., CM/SEC AL SEC., FT/SEC AL SEC, KM/H AL SEC.
1160 RESTORE 1150
1170 GOSUB 2840
1180 DATA 1,1E+2,3.281,3.6
1190 DATA 1E-2,1.3,281E-2,3.6E-2
1200 DATA .3048,30.48,1.1,0.97
1210 DATA .2778,27.778,1.91,1
1220 GOSUB 2880
1230 GOSUB 2720
1240 GOSUB 2600
1250 GOSUB 2780
1260 GOSUB 2470
1270 IF K=83 THEN 1230 ELSE 270
1280 CALL CLEAR
1290 PRINT "MASSA":
1300 GOSUB 2690
1310 MAX=4
1320 DATA GRAMMO, KG, ONCIA, LIBBRA
1330 RESTORE 1320
1340 GOSUB 2840

```

```

1350 DATA 1,1E-3,3.527E-2,2.2E-3
1360 DATA 1000,1,35.27,2.204
1370 DATA 28.349,2.835E-2,1,6.25E-2
1380 DATA 453.592,.4536,16,1
1390 GOSUB 2880
1400 GOSUB 2720
1410 GOSUB 2600
1420 GOSUB 2780
1430 GOSUB 2470
1440 IF K=83 THEN 1400 ELSE 270
1450 CALL CLEAR
1460 PRINT "FORZA":
1470 GOSUB 2690
1480 MAX=4
1490 DATA DINA,NEWTON,KGF,LBF
1500 RESTORE 1490
1510 GOSUB 2840
1520 DATA 1,1E-5,1.02E-6,2.248E-6
1530 DATA 1E+5,1,102,2.25
1540 DATA 9.8E+5,9.8,1,2.2
1550 DATA 4.445E-5,4.445,.454,1
1560 GOSUB 2880
1570 GOSUB 2720
1580 GOSUB 2600
1590 GOSUB 2780
1600 GOSUB 2470
1610 IF K=83 THEN 1570 ELSE 270
1620 CALL CLEAR
1630 PRINT "PRESSIONE":
1640 GOSUB 2690
1650 MAX=5
1660 DATA DINE/CMQ,NEWTON/MQ,LBF/INQ,ATMOSFERA,BAR
1670 RESTORE 1660
1680 GOSUB 2840
1690 DATA 1,1,1.45E-5,9.265E-7,1E-6
1700 DATA 10,1,1.45E-4,9.871E-6,1E-5
1710 DATA 6.896E+4,6.896E+3,1,.068,.0688
1720 DATA 1.013E+6,1.013E+5,14.7,1,1.013
1730 DATA 1E+6,1E+5,14.5,.9871,1
1740 GOSUB 2880
1750 GOSUB 2720
1760 GOSUB 2600
1770 GOSUB 2780
1780 GOSUB 2470
1790 IF K=83 THEN 1750 ELSE 270
1800 CALL CLEAR
1810 PRINT "ENERGIA":
1820 GOSUB 2690
1830 MAX=6
1840 DATA ERG,JOULE,KGM,CAL,KWH,EV
1850 RESTORE 1840
1860 GOSUB 2840
1870 DATA 1,1E-7,1.02E-8,2.39E-9,2.78E-14,6.29E-11
1880 DATA 1E+7,1,102,2.39,2.78E-5,6.29E-18
1890 DATA 9.8E+7,9.8,1,2.343,2.72E-6,6.2E+19
1900 DATA 4.186E+7,4.186,4243,1,1.18E-6,2.6E+19
1910 DATA 3.6E+13,3.6E+6,3.63E+5,8.6E+5,1,2.3E+25
1920 DATA 1.59E-12,1.59E-19,1.62E-20,3.7E-20,4.4E-26,1
1930 GOSUB 2880
1940 GOSUB 2720
1950 GOSUB 2600
1960 GOSUB 2780
1970 GOSUB 2470
1980 IF K=83 THEN 1940 ELSE 270
1990 CALL CLEAR
2000 PRINT "POTENZA":
2010 GOSUB 2690
2020 MAX=4
2030 DATA ERG/SEC,WATT,HP,CAL/SEC
2040 RESTORE 2030
2050 GOSUB 2840
2060 DATA 1,1E-7,1.34E-10,2.39E-8
2070 DATA 1E+7,1,1.34E-3,2.39
2080 DATA 7.35E+9,735.5,1,178
2090 DATA 4.186E+7,4.186,3.62E-3,1
2100 GOSUB 2880
2110 GOSUB 2720
2120 GOSUB 2600
2130 GOSUB 2780
2140 GOSUB 2470
2150 IF K=83 THEN 2110 ELSE 270
2160 CALL CLEAR
2170 PRINT "TEMPERATURE":
2180 GOSUB 2690
2190 MAX=3
2200 DATA GRADI KELVIN,GRADI CELSIUS,GRADI FAHRENHEIT

```

```

2210 RESTORE 2200
2220 GOSUB 2840
2230 GOSUB 2720
2240 GOSUB 2600
2250 PRINT A$(U1);
2260 INPUT X
2270 PRINT
2280 ON U1 GOTO 2290,2350,2400
2290 IF U2=3 THEN 2320
2300 R=X-273.1
2310 GOTO 2440
2320 R=X-273.1
2330 R=(9/5)*R+32
2340 GOTO 2440
2350 IF U2=1 THEN 2380
2360 R=(9/5)*X+32
2370 GOTO 2440
2380 R=X+273.1
2390 GOTO 2440
2400 IF U2=1 THEN 2430
2410 R=(5/9)*(X-32)
2420 GOTO 2440
2430 R=(5/9)*(X-32)+273.1
2440 PRINT " = " | A$(U2); R
2450 GOSUB 2470
2460 IF K=83 THEN 2230 ELSE 270
2470 PRINT " : "
2480 CALL HCHAR(22,3,128,28)
2490 PRINT " - VUOI CONTINUARE? (S/N) - "
2500 CALL KEY(0,K,S)
2510 IF S=0 THEN 2500
2520 IF (K<>78)*(K<>83) THEN 2500
2530 CALL CLEAR
2540 RETURN
2550 CALL KEY(0,K,S)
2560 IF S=0 THEN 2550
2570 K=K-48
2580 IF (K<1)+(K>MAX) THEN 2550
2590 RETURN
2600 PRINT "SCEGLI L'UNITA' DI MISURA"
2610 PRINT "NELLA QUALE E' ESPRESSA LA", "GRANDEZZA..." :
2620 GOSUB 2550
2630 UI=K
2640 PRINT " ...E L'UNITA' DI MISURA"
2650 PRINT "NELLA QUALE DEVE ESSERE", "CONVERTITA." :
2660 GOSUB 2550
2670 U2=K
2680 RETURN
2690 CALL HCHAR(24,3,128,28)
2700 PRINT " : "
2710 RETURN
2720 PRINT "UNITA' DI MISURA":
2730 FOR I=1 TO MAX
2740 PRINT STR$(I); ". " | A$(I)
2750 NEXT I
2760 PRINT
2770 RETURN
2780 PRINT A$(U1);
2790 INPUT X
2800 PRINT
2810 R=INT(1000*X*C(U1,U2))/1000
2820 PRINT " = " | A$(U2); R
2830 RETURN
2840 FOR I=1 TO MAX
2850 READ A$(I)
2860 NEXT I
2870 RETURN
2880 FOR I=1 TO MAX
2890 FOR J=1 TO MAX
2900 READ C(I,J)
2910 NEXT J
2920 NEXT I
2930 RETURN

```

Di tutti i programmi del libro questo presenta una struttura meno differenziata nelle sue parti. Per questa ragione sarà sufficiente esaminare in dettaglio una sola sezione del programma. Prendiamo ad esempio quella che converte le unità di misura delle lunghezze; la troviamo alle righe 560-760.

Dopo il sottotitolo LUNGHEZZA, l'istruzione GOSUB di linea 580 ri-

chiama la routine per tracciare una linea orizzontale ed uno spazio. L'istruzione DATA contiene le unità di misura che possono essere convertite una nell'altra, il loro numero è posto in MAX (linea 610). Il sottoprogramma 2840 legge i sette dati e li carica nel vettore A\$(1).

Le righe 630-690 contengono la tabella dei coefficienti per le conversioni; essa viene memorizzata nella matrice bidimensionale C(I,J) tramite il sottoprogramma 2880.

La routine 2720 stampa sul video le unità di misura, mentre la 2600 riceve i dati in input; l'unità nella quale è espressa la misura è associata alla variabile U1, quella nella quale deve essere convertita è associata alla variabile U2. Ancora una routine, la 2760, chiede la misura X da convertire dopo aver stampato l'unità A\$(U1) e calcola il risultato moltiplicando X per il coefficiente C(U1,U2) e lo stampa.

La GOSUB di riga 740 provoca l'apparizione della richiesta se si vuole continuare con lo stesso tipo di conversioni; in caso affermativo si rimane nella stessa sezione con una nuova richiesta di dati in input, altrimenti si ritorna al menù principale.

Come si vede, ogni parte del programma contiene essenzialmente la tabella delle unità e quella dei coefficienti di conversione, tutto il resto è svolto da una serie di routine. Il risparmio di memoria è evidente.

Il programma occupa 6.6 Kbyte.

Tra l'utile e il dilettevole

Dopo tanti programmi di matematica, alcuni dei quali di una certa complessità, un po' di distensione. Il TI 99/4A oltre ad essere un buon computer per risolvere problemi scientifici ha delle prerogative che lo rendono adatto ad una grande varietà di applicazioni che questo libro, per le sue finalità, non ha potuto prendere in considerazione.

Vogliamo sopperire almeno in parte con i programmi presentati in quest'ultimo capitolo.

Ci sono almeno due caratteristiche che fino ad ora non abbiamo utilizzato: la capacità di generare suoni e la possibilità di memorizzare dati su un nastro magnetico per richiamarli in seguito a seconda della necessità. La musica non era ancora stata inserita per non ampliare troppo i programmi e non rallentare la loro esecuzione; per di più, nella maggior parte dei casi sarebbe stata del tutto fuori luogo.

Nei programmi che seguono, alcuni motivetti rendono più piacevole la presentazione o costituiscono una sorta di premio dopo una serie di risposte esatte.

Un programma utilizza file sequenziali su nastro. Poiché i comuni programmi che memorizzano indirizzi e numeri telefonici si rivelano inutili al lato pratico in quanto non permettono un accesso diretto ai record, si è scelta un'applicazione originale che costituisce un supporto ad un gioco molto diffuso: quello degli scacchi. Il programma permette la costituzione di un archivio di aperture o di intere partite che vengono poi riprodotte su una scacchiera che appare sul video accompagnata anche dalla notazione scritta delle mosse.

Bisogna ricordare che la capacità di memorizzare dati sotto forma di record è uno dei motivi essenziali che hanno portato alla creazione degli elaboratori.

L'ultimo programma è un accenno ad un campo applicativo che va sotto il nome ambizioso di intelligenza artificiale.

Il computer, partendo da una situazione reale, può indagare gli sviluppi futuri dovuti al cambiamento di alcuni parametri; se opportunamente programmato può acquisire una strategia vincente e competere alla pari con l'uomo in molti giochi.

Passiamo ora ai programmi.

Città d'Italia

L'argomento, come fa presagire il titolo, è di natura geografica anche se dalla geografia prende a prestito solo elementi di tipo mnemonico. Pur essendo sotto certi aspetti di tipo educativo, il programma è semplicemente un gioco presentato sotto forma di quiz.

Si tratta di indovinare il nome delle città che appaiono casualmente su una cartina "muta" d'Italia. Possono giocare fino a cinque concorrenti a ciascuno dei quali vengono rivolte 10 domande. Un totalizzatore delle risposte esatte tiene aggiornata la situazione.

Eccome come appare il programma in esecuzione.

RUN

```

CONCORRENTE B

SCRIVI IL NOME
DELLA CITTA'
ROMA                                <ENTER>
-----
                                ESATTO
•

CONCORRENTI:  A  B
PUNTI: 10      B
  
```

Il programma si presta facilmente ad essere cambiato perché le domande riguardino anche altre città, oltre le 20 già memorizzate, o addirittura al-

tri elementi come monti, fiumi, laghi, ecc. In questi ultimi casi sarebbe opportuno cambiare il simbolo che li deve rappresentare e che attualmente è un piccolo cerchio.

Vediamo ora il listado.

```

100 REM *****
110 REM *
120 REM * CITTA' *
130 REM * D'ITALIA *
140 REM *
150 REM *****
160 REM di: Sergio Barmani
170 REM versione:
180 REM TI EXTENDED BASIC.
190 CALL CLEAR
200 CALL CHAR(96,RPT$( "F",16))
210 CALL CHAR(97,RPT$( "O",12)&"0103")
220 CALL CHAR(98,RPT$( "O",9)&"F3FFFF")
230 CALL CHAR(99,"00000018F8FCFCFE")
240 CALL CHAR(100,"COFB"&RPT$( "F",12))
250 CALL CHAR(101,"0000COFBFOEOFOFB")
260 CALL CHAR(102,"00002040EOFOFEFE")
270 CALL CHAR(103,"000038387F7FFF")
280 CALL CHAR(104,"030B1F1F1F9FDFFF")
290 CALL CHAR(105,"070FOFOFO7070303")
300 CALL CHAR(106,RPT$( "F",13)&"9E")
310 CALL CHAR(107,"FBFOFOFBFCFCFC38")
320 CALL CHAR(108,"03030F1F3F3F0707")
330 CALL CHAR(109,"8000000000808080")
340 CALL CHAR(110,"1F1FOFOFO7070303")
350 CALL CHAR(111,"FEFCFCFCFCFCFCFE")
360 CALL CHAR(112,RPT$( "F",14)&"3F")
370 CALL CHAR(113,"3F3E7C78FBFOCO")
380 CALL CHAR(114,"1FOFO703")
390 CALL CHAR(115,"FFFFFFFF7F7F3F3F")
400 CALL CHAR(116,"0080COEOFOFBFCFF")
410 CALL CHAR(117,"3F1F1F1F1FOFOFOF")
420 CALL CHAR(118,"COFOFOFBFBFCFC")
430 CALL CHAR(119,"OFOFO101FO70")
440 CALL CHAR(120,"FCFCFEFEFEFEFFFF")
450 CALL CHAR(121,"FFFFFFFF7F3F1F")
460 CALL CHAR(122,"0703")
470 CALL CHAR(123,"FFFF3FO703")
480 CALL CHAR(124,"OFOFO8")
490 CALL CHAR(125,"00000000000407.7")
500 CALL CHAR(126,"0001071F3F7FFFFF")
510 CALL CHAR(127,"EOEOEOEOFOFOFBFB")
520 CALL CHAR(128,"0707"&RPT$( "01",6))
530 CALL CHAR(129,"FBFCFCFBFBFOFOFO")
540 CALL CHAR(130,"FBFBFBEOEOFOFBFB")
550 CALL CHAR(131,"0000080COEOFOFBFB")
560 CALL CHAR(132,"FFFF7F3F3F7FFF7F")
570 CALL CHAR(133,"FFFFFFFF7F3F03")
580 CALL CHAR(134,"1FO6")
590 CALL CHAR(135,"FF7F7F3F3F3F3F3F")
600 CALL CHAR(136,"COCOCOCOCOB0COCFO")
610 CALL CHAR(137,"7C7C3838")
620 CALL CHAR(138,"0303030F1FOFOF1F")
630 CALL CHAR(139,"FBFOEOEOEOFOFOFO")
640 CALL CHAR(140,"1F3F3F7F7F3E1C08")
650 CALL CHAR(141,"EOC080")
660 CALL CHAR(142,"0303030101010303")
670 CALL CHAR(143,"FOFOFOFOEOEOEOEO")
680 CALL CHAR(144,"0303070707010303")
690 CALL CHAR(145,"FFFFFFFFFBFOEOCO")
700 CALL CHAR(146,"EOEOEOEOEOEOCO")
710 CALL CHAR(147,"0000000163F3FFFF")
720 CALL CHAR(148,"000000COCOFFFFFF")
730 CALL CHAR(149,"0000001FFFFFFFFE")
740 CALL CHAR(150,"FFFFFFFFFEFC1C")
750 CALL CHAR(151,"0000183C3C18")
760 CALL CHAR(152,"00201008")
770 CALL SCREEN(8)
780 CALL COLOR(13,1)
790 FOR C=9 TO 14: CALL COLOR(C,13,1): NEXT C
800 CALL HCHAR(3,9,97): CALL HCHAR(3,10,98): CALL HCHAR(3,11,99)
810 CALL HCHAR(4,6,97): CALL HCHAR(4,7,102): CALL HCHAR(4,8,103)
820 CALL HCHAR(4,9,104): CALL HCHAR(4,10,96,2): CALL HCHAR(4,12,100)
830 CALL HCHAR(4,13,101): CALL HCHAR(5,5,105): CALL HCHAR(5,6,96,6)
840 CALL HCHAR(5,12,106): CALL HCHAR(5,13,107): CALL HCHAR(5,5,108)
850 CALL HCHAR(6,6,96,6): CALL HCHAR(6,12,109): CALL HCHAR(7,5,110)

```



```

860 CALL HCHAR(7,6,96):: CALL HCHAR(7,7,106):: CALL HCHAR(7,8,112)
870 CALL HCHAR(7,9,96,2):: CALL HCHAR(7,11,111):: CALL HCHAR(8,6,113)
880 CALL HCHAR(8,8,114):: CALL HCHAR(8,9,115):: CALL HCHAR(8,10,96,2)
890 CALL HCHAR(8,12,116):: CALL HCHAR(9,8,117):: CALL HCHAR(9,10,96,3)
900 CALL HCHAR(9,13,118):: CALL HCHAR(10,9,119):: CALL HCHAR(10,10,115)
910 CALL HCHAR(10,11,96,2):: CALL HCHAR(10,13,120):: CALL HCHAR(11,10,114)
920 CALL HCHAR(11,11,121):: CALL HCHAR(11,12,96,2):: CALL HCHAR(11,14,116)
930 CALL HCHAR(12,8,114):: CALL HCHAR(12,12,121):: CALL HCHAR(12,13,96,3)
940 CALL HCHAR(12,14,130):: CALL HCHAR(13,12,125):: CALL HCHAR(13,7,126)
950 CALL HCHAR(13,8,127):: CALL HCHAR(13,12,122):: CALL HCHAR(13,13,123)
960 CALL HCHAR(13,14,96,3):: CALL HCHAR(13,17,100):: CALL HCHAR(13,18,131)
970 CALL HCHAR(14,6,128):: CALL HCHAR(14,7,96):: CALL HCHAR(14,8,129)
980 CALL HCHAR(14,14,124):: CALL HCHAR(14,15,132):: CALL HCHAR(14,16,96)
990 CALL HCHAR(14,17,34):: CALL HCHAR(14,18,133):: CALL HCHAR(14,19,118)
1000 CALL HCHAR(15,6,142):: CALL HCHAR(15,7,96):: CALL HCHAR(15,8,143)
1010 CALL HCHAR(15,13,134):: CALL HCHAR(15,16,135):: CALL HCHAR(15,17,136)
1020 CALL HCHAR(15,19,137)
1030 CALL HCHAR(16,6,33):: CALL HCHAR(16,7,34):: CALL HCHAR(16,8,35)
1040 CALL HCHAR(16,11,110):: CALL HCHAR(16,17,96)
1050 CALL HCHAR(17,16,138):: CALL HCHAR(17,17,139)
1060 CALL HCHAR(18,12,36):: CALL HCHAR(18,13,37):: CALL HCHAR(18,14,98)
1070 CALL HCHAR(18,15,38):: CALL HCHAR(18,16,140):: CALL HCHAR(18,17,141)
1080 CALL HCHAR(19,2,123):: CALL HCHAR(19,13,121):: CALL HCHAR(19,14,96)
1090 CALL HCHAR(19,15,111):: CALL HCHAR(20,13,122)
1100 CALL HCHAR(20,14,125):: CALL HCHAR(20,15,39)
1110 DISPLAY AT(16,16) "CITTA";CHR$(47):: DISPLAY AT(8,16) "ITALIANE"
1120 CALL SOUND(150,659,2)
1130 CALL SOUND(150,657,2)
1140 CALL SOUND(300,523,2)
1150 CALL SOUND(300,440,2)
1160 CALL SOUND(300,698,2,440,5)
1170 CALL SOUND(300,784,2,587,5)
1180 CALL SOUND(150,698,2,392,5)
1190 CALL SOUND(150,659,2)
1200 CALL SOUND(1000,698,2,440,5)
1210 DIM A$(20,3)
1220 DATA ROMA,86,86,MILANO,38,56,TORINO,42,40,VENEZIA,39,86,BOLOGNA,50,74
1230 DATA TRIESTE,36,93,NAPOLI,100,105,PALERMO,139,96,CASLIARI,120,50
1240 DATA GENOVA,50,51,PERUGIA,71,85,ANCONA,63,96,BARI,95,129
1250 DATA PISA,62,64,ADOSTA,33,37,BOLZANO,23,75,PARANZO,105,132
1260 DATA PADOVA,39,80,FIRENZE,62,74,TRENTO,29,72
1270 FOR I=1 TO 20: READ A$(I,1),A$(I,2),A$(I,3):: NEXT I
1280 DISPLAY AT(22,2) "QUANTI CONCORRENTI?"
1290 DISPLAY AT(24,3) " ( AL MASSIMO 5 ) "
1300 ACCEPT AT(24,24) BEEP VALIDATE(DIGIT)::N
1310 IF N<1 OR N>5 THEN 1300
1320 CALL HCHAR(22,1,32,32)
1330 CALL HCHAR(24,1,32,32)
1340 FOR J=1 TO N: P(J)=0: NEXT J
1350 DISPLAY AT(22,2) "CONCORRENTI: " : DISPLAY AT(24,8) "PUNTI: "
1360 FOR K=1 TO N: DISPLAY AT(22,13+3*K) CHR$(64+K)
1370 DISPLAY AT(24,12+3*K) USING "##";P(K):: NEXT K
1380 CALL HCHAR(6,16,32,14)
1390 RANDOMIZE
1400 FOR I=1 TO N
1410 DISPLAY AT(1,7) BEEP "CONCORRENTE" : DISPLAY AT(1,19) CHR$(64+I)
1420 DISPLAY AT(5,14) "SCRIVI IL NOME"
1430 DISPLAY AT(6,14) "DELLA CITTA" CHR$(47)
1440 CALL HCHAR(9,16,95,11)
1450 CALL SPRITE(61,43,2,80,120)
1460 FOR K=1 TO 10
1470 R=INT(20*RND)+1
1480 X=VAL(A$(R,2))
1490 Y=VAL(A$(R,3))
1500 CALL LOCATE(81,X,Y)
1510 ACCEPT AT(8,14) BEEP VALIDATE(ULPHA):RISP$
1520 IF RISP$<>A$(R,1) THEN 1560
1530 DISPLAY AT(12,19) "ESATTO"
1540 P(I)=P(I)+1
1550 GOTO 1570
1560 DISPLAY AT(12,19) "SBAGLIATO"
1570 DISPLAY AT(24,12+3*I) BEEP SIZE(2) USING "##";P(I)
1580 FOR T=1 TO 500: NEXT T
1590 CALL HCHAR(8,16,32,14):: CALL HCHAR(12,18,32,12)
1600 NEXT K
1610 CALL SOUND(225,784,2)
1620 CALL SOUND(150,659,2)
1630 CALL SOUND(300,740,2)
1640 CALL SOUND(300,587,2)
1650 CALL SOUND(300,494,2)
1660 CALL SOUND(300,784,2)
1670 CALL SOUND(1000,659,2)
1680 CALL SOUND(300,587,2)
1690 CALL SOUND(300,587,2)
1700 CALL SOUND(300,587,2)
1710 NEXT I

```

```

1720 DISPLAY AT(1,1)BEEP:"* VUOI CONTINUARE?"
1730 ACCEPT AT(1,2)VALIDATE(UALPHA):RR$
1740 IF SEG$(RR$,1,1)="S" THEN 1800
1750 DISPLAY AT(8,14):" FINE"
1760 FOR T=1 TO 1000:: NEXT T
1770 CALL DELSPRITE(81)
1780 CALL DELSPRITE(81)
1790 CALL CLEAR:: STOP
1800 DISPLAY AT(1,1):" "
1810 FOR RIGA=4 TO 9:: DISPLAY AT(RIGA,14):" " :: NEXT RIGA
1820 DISPLAY AT(12,16):" "
1830 FOR RIGA=22 TO 24 STEP 2:: DISPLAY AT(RIGA,1):" " :: NEXT RIGA
1840 CALL DELSPRITE(81)
1850 BOTO 1280
1860 END

```

Alle righe 200-760 sono definiti i caratteri grafici per disegnare la cartina d'Italia ad alta risoluzione. L'Extended Basic non può utilizzare i caratteri degli ultimi due sottoinsiemi numerici, il 15 ed il 16. Dato l'elevato numero di caratteri necessari si è dovuto ridefinire molti caratteri ASCII.

Tutte le istruzioni CALL HCHAR dalla 800 alla 1100, servono a disegnare l'Italia. Seguono il titolo e alcune note di accompagnamento. Le istruzioni DATA alle righe 1220-1260 contengono i nomi delle città e le loro coordinate video in pixel; tutti questi elementi vengono memorizzati nella matrice A\$. N è il numero dei concorrenti (riga 1300); essi vengono denominati con le lettere dell'alfabeto, A, B, C, D, E, e, in successione, sono invitati a rispondere alla riga 1410.

Lo sprite di riga 1450 è il tondino che rappresenta la città; in base alla funzione RND viene posto nel punto video di coordinate X,Y (linea 1500). Se la risposta RISP\$ è esatta viene incrementato il contatore P(I) con l'istruzione 1540 e compare il messaggio ESATTO.

Nulla impedisce che nel corso delle dieci domande debba ripetersi la stessa città due o più volte, la loro scelta è casuale e non vengono memorizzate le domande già fatte.

Quando un concorrente ha terminato viene eseguita la musica con le istruzioni 1610-1700.

Se si ricomincia, è necessario azzerare il contatore P(I); l'operazione è svolta alla riga 1340.

Archivio scacchistico

Chiariamo subito che il programma non consente di giocare a scacchi (esiste a questo scopo un modulo Solid State della Texas Instruments) ma consente di memorizzare partite come quelle che vengono pubblicate in alcune riviste e nei manuali specializzati. Le partite vengono conservate sotto forma di file sul nastro magnetico ed in seguito possono essere richiamate ed eseguite sulla scacchiera che appare sul video.

Il programma si divide in due parti: una per registrare una partita, un'altra per richiamarla ed eseguirla.

La prima sezione comprende anche le istruzioni per inserire correttamente i dati, tuttavia, poiché una loro errata trascrizione può compromettere il funzionamento del programma, è preferibile seguire con attenzione le seguenti indicazioni.

I primi dati consistono nei nomi dei giocatori, nel luogo e nella data in cui si è svolta la partita ed infine nel tipo di apertura. Luogo e data formano un'unica stringa e non devono essere separate da una virgola perché questo segno di interpunzione serve a separare più variabili in input. Tutti i dati sopra menzionati formano il primo record del file, i record successivi contengono le mosse che formano la partita. Ogni mossa completa occupa una stringa di 12 caratteri, i primi 6 per la mossa del bianco, gli ultimi 6 per la mossa del nero. Ogni record riunisce quattro mosse complete e comprende pertanto quattro stringhe di 12 caratteri ciascuna.



Per il lettore riveste maggior interesse la trascrizione delle mosse. Il numero progressivo appare automaticamente e non deve essere trascritto; bisogna invece indicare necessariamente il pezzo bianco che muove (sempre con lettere maiuscole) usando i simboli convenzionali: P = pedone, C = cavallo, A = alfiere, T = torre, D = dama, R = re. Il secondo ed il terzo carattere contengono le coordinate della casa di partenza mentre il quarto ed il quinto carattere si riferiscono alla casa di arrivo. Il sesto carattere è normalmente un trattino ma può essere il segno "+" per indicare uno scacco al re, il segno ":" per indicare la cattura di un pezzo, il segno "!" per indicare un'ottima mossa, il segno "%" per la presa *en passant*, e ancora "=" per lo scacco matto; se c'è la promozione di un pezzo questo stesso sesto carattere deve essere necessariamente il simbolo

di un pezzo: C, A, T, D. Ovviamente non si può ottenere un re con la promozione. Per le mosse del nero valgono le stesse regole relativamente ai caratteri dal settimo al dodicesimo.

Quando termina la trascrizione delle mosse è indispensabile che il dodicesimo carattere della stringa sia un punto perché così il computer riconosce che la partita è finita.

Resta ancora da definire l'arrocco; si usino i seguenti simboli: -0-0- per l'arrocco corto e 0-0-0 per quello lungo, occupando i primi cinque caratteri di ogni semistringa.

Come si vede la notazione scritta delle mosse è quasi uguale a quella convenzionale, solo che quest'ultima è generalmente usata in una forma semplificata.

Un esempio concreto chiarirà ulteriormente come avviene l'introduzione dei dati.

Quello che segue è il classico scacco matto di Légal.

1. e4, e5;	PE2E4-PE7E5-
2. Cf3, Cc6;	CG1F3-CB8C6-
3. Ac4, d6;	AF1C4-PD7D6-
4. Cc3, Ag4;	CB1C3-AC8G4-
5. C:e5, A:Dd1;	CF3E5:AG4D1:
6. A:f7+, Re7;	AC4F7+REBE7-
7. Cc3-d5 =	CC3D5=MATTO.

Come si vede si può inserire anche un commento, infatti il computer vedendo il punto al sesto posto della semistringa non eseguirà alcuna mossa per il nero.

Ecco ora il listato del programma; esso occupa 6.6 Kbyte di memoria.

```

100 REM *****
110 REM *
120 REM *   ARCHIVIO   *
130 REM * SCACCHISTICO *
140 REM *
150 REM *****
160 REM di: Sergio Borgani
170 REM versione: TI BASIC
180 CALL CLEAR
190 DIM A$(100)
200 PRINT TAB(11);"ARCHIVIO":;
210 PRINT TAB(9);"SCACCHISTICO":;:;:;
220 PRINT "PREMI: ";
230 PRINT "1.: REGISTRARE UNA PARTITA":;
240 PRINT "2.: RICHIAMARE UNA PARTITA":;
250 PRINT "3.: STOP":;
260 INPUT "SCELTA? ":"S$
270 IF S$="1" THEN 320
280 IF S$="2" THEN 770
290 IF S$="3" THEN 2790
300 GOTO 260
310 REM *** SAVE ***
320 CALL CLEAR
330 PRINT "REGISTRARE UNA PARTITA":;
340 PRINT "SCRIVI I NOMI DEI GIOCATORI"
350 INPUT "BIANCO ":"G1$
360 INPUT "NERO ":"G2$
370 PRINT
380 PRINT "LUOGO E DATA"
```

```

390 INPUT LD$
400 PRINT
410 INPUT "APERTURA: " ; AP$
420 PRINT
430 N=0
440 PRINT "SCRIVI LE MOSSE UNA AD UNA"
450 PRINT "ES.: PE2E4-PC7C5-""
460 PRINT "PER FINIRE TERMINA CON UN PUNTO:."
470 PRINT "AL PASSAGGIO: X""
480 PRINT "PROMOZIONE: D,T,A, OPPURE C""
490 PRINT "ARROCCO: -0-0- OPPURE 0-0-0""
500 PRINT TAB(9); "-----"
510 N=N+1
520 IF N<101 THEN 560
530 PRINT "MEMORIA FULL"
540 CALL SOUND(1000,440,2)
550 GOTO 180
560 PRINT TAB(6-LEN(STR$(N))); STR$(N); ". "
570 INPUT AS(N)
580 CALL MCHAR(23,9,32)
590 IF LEN(AS(N))=12 THEN 620
600 PRINT "ERRORE! RIPROVA"
610 GOTO 560
620 IF SEG$(AS(N),12,1)=". " THEN 640
630 GOTO 510
640 CALL CLEAR
650 INPUT "VUOI REGISTRARE? (S/N) " ; RS$
660 IF SEG$(RS$,1,1)="S" THEN 680
670 GOTO 320
680 OPEN #1:"CS1",SEQUENTIAL,OUTPUT,INTERNAL,FIXED
690 K=1
700 PRINT #1;G1$;G2$;LD$;AP$
710 PRINT #1;AS(K);AS(K+1);AS(K+2);AS(K+3)
720 K=K+4
730 IF K>N THEN 750
740 GOTO 710
750 CLOSE #1
760 GOTO 180
770 CALL CLEAR
780 RE$="081C087F493E1C00"
790 DA$="000049497F7F3E00"
800 AL$="000818323E087F00"
810 CA$="107C7E0E1E0C3E00"
820 TO$="002A3E1C1C1C3E00"
830 PE$="00000808081C3E00"
840 CALL CHAR(128,RE$)
850 CALL CHAR(129,DA$)
860 CALL CHAR(130,AL$)
870 CALL CHAR(131,CA$)
880 CALL CHAR(132,TO$)
890 CALL CHAR(133,PE$)
900 CALL CHAR(134,"0")
910 CALL CHAR(136,RE$)
920 CALL CHAR(137,DA$)
930 CALL CHAR(138,AL$)
940 CALL CHAR(139,CA$)
950 CALL CHAR(140,TO$)
960 CALL CHAR(141,PE$)
970 CALL CHAR(144,RE$)
980 CALL CHAR(145,DA$)
990 CALL CHAR(146,AL$)
1000 CALL CHAR(147,CA$)
1010 CALL CHAR(148,TO$)
1020 CALL CHAR(149,PE$)
1030 CALL CHAR(152,RE$)
1040 CALL CHAR(153,DA$)
1050 CALL CHAR(154,AL$)
1060 CALL CHAR(155,CA$)
1070 CALL CHAR(156,TO$)
1080 CALL CHAR(157,PE$)
1090 CALL CHAR(158,"0")
1100 CALL CHAR(134,"00000000000000FF")
1110 CALL CHAR(136,"01010101010101")
1120 CALL CHAR(144,"0808080808080808")
1130 CALL CHAR(138,"FF")
1140 CALL COLOR(13,16,6)
1150 CALL COLOR(14,16,11)
1160 CALL COLOR(15,2,6)
1170 CALL COLOR(16,2,11)
1180 YY(1)=129
1190 YY(2)=132
1200 YY(3)=130
1210 YY(4)=131
1220 REM *** OLD ***
1230 OPEN #1:"CS1",INPUT,SEQUENTIAL,INTERNAL,FIXED
1240 K=1
1250 INPUT #1;G1$,G2$,LD$,AP$

```

```

1260 INPUT #1:A$(K),A$(K+1),A$(K+2),A$(K+3)
1270 FOR J=K TO K+3
1280 IF SEG$(A$(J),12,1)=". " THEN 1320
1290 NEXT J
1300 K=K+4
1310 GOTO 1260
1320 CLOSE #1
1330 CALL CLEAR
1340 PRINT TAB(14-LEN(LD$)/2);LD$::
1350 PRINT TAB(14-LEN(AP$)/2);AP$::
1360 PRINT TAB(3);G2$::
1370 PRINT TAB(3);G1$::
1380 PRINT " * PREMI UN TASTO *"
1390 CALL HCHAR(10,4,34,8)
1400 CALL VCHAR(11,4,34,8)
1410 CALL VCHAR(11,13,64,8)
1420 CALL HCHAR(19,5,38,8)
1430 DATA 156,147,154,145,152,146,155,148
1440 RESTORE 1430
1450 FOR J=1 TO 8
1460 READ PEZZO
1470 CALL HCHAR(11,4+J,PEZZO)
1480 NEXT J
1490 FOR J=1 TO 8 STEP 2
1500 CALL HCHAR(12,4+J,149)
1510 CALL HCHAR(12,5+J,137)
1520 NEXT J
1530 Q=12
1540 FOR RIGA=13 TO 16
1550 FOR J=1 TO 8 STEP 2
1560 CALL HCHAR(RIGA,4+J,146+Q)
1570 CALL HCHAR(RIGA,5+J,146-Q)
1580 NEXT J
1590 Q=-Q
1600 NEXT RIGA
1610 FOR J=1 TO 8 STEP 2
1620 CALL HCHAR(17,4+J,141)
1630 CALL HCHAR(17,5+J,133)
1640 NEXT J
1650 DATA 132,139,130,137,128,138,131,140
1660 RESTORE 1650
1670 FOR J=1 TO 8
1680 READ PEZZO
1690 CALL HCHAR(18,4+J,PEZZO)
1700 NEXT J
1710 N=0
1720 CALL KEY(0,KEY,STATUS)
1730 IF STATUS=0 THEN 1720
1740 N=N+1
1750 REM *** STAMPA RIGA ***
1760 M=N
1770 FOR RIGA=20 TO 9 STEP -1
1780 RIG$=STR$(N)&" "&A$(M)
1790 IF M<1 THEN 1870
1800 FOR J=1 TO LEN(RIG$)
1810 L=ASC(SEG$(RIG$,J,1))
1820 CALL HCHAR(RIGA,18+J,L)
1830 NEXT J
1840 M=M-1
1850 NEXT RIGA
1860 REM *** MOSSA ***
1870 M=1
1880 IF M=1 THEN 1900
1890 M=7
1900 MOSSA$=SEG$(A$(N),M,6)
1910 IF SEG$(MOSSA$,1,5)="-0-0-" THEN 2250
1920 IF SEG$(MOSSA$,1,5)="-0-0-0-" THEN 2360
1930 IF SEG$(MOSSA$,6,1)="-." THEN 2740
1940 R1=19-VAL(SEG$(MOSSA$,3,1))
1950 C1=ASC(SEG$(MOSSA$,2,1))-60
1960 R2=19-VAL(SEG$(MOSSA$,5,1))
1970 C2=ASC(SEG$(MOSSA$,4,1))-60
1980 CALL GCHAR(R1,C1,PX)
1990 IF PX>M*3+130 THEN 2060
2000 VECCHIO=134
2010 IF (R2+C2)/2=INT((R2+C2)/2) THEN 2040
2020 PEZZO=PX
2030 GOTO 2110
2040 PEZZO=PX+8
2050 GOTO 2110
2060 VECCHIO=138
2070 IF (R2+C2)/2=INT((R2+C2)/2) THEN 2100
2080 PEZZO=PX-8
2090 GOTO 2110
2100 PEZZO=PX
2110 CALL HCHAR(R1,C1,VECCHIO)

```

```

2120 CALL HCHAR(R2,C2,PEZZO)
2130 FF$=SEG$(MOSSA$,5,1)
2140 IF FF$="" THEN 2480
2150 PRO=1
2160 IF FF$="D" THEN 2600
2170 PRO=2
2180 IF FF$="T" THEN 2600
2190 PRO=3
2200 IF FF$="A" THEN 2600
2210 PRO=4
2220 IF FF$="C" THEN 2600
2230 IF M=1 THEN 1890 ELSE 1720
2240 REM *** ARROCCO ***
2250 IF M=1 THEN 2310
2260 CALL HCHAR(11,12,134)
2270 CALL HCHAR(11,10,148)
2280 CALL HCHAR(11,9,158)
2290 CALL HCHAR(11,11,152)
2300 GOTO 2230
2310 CALL HCHAR(18,9,134)
2320 CALL HCHAR(18,11,128)
2330 CALL HCHAR(18,12,138)
2340 CALL HCHAR(18,10,140)
2350 GOTO 2230
2360 IF M=1 THEN 2420
2370 CALL HCHAR(11,9,158)
2380 CALL HCHAR(11,7,152)
2390 CALL HCHAR(11,5,158)
2400 CALL HCHAR(11,8,148)
2410 GOTO 2230
2420 CALL HCHAR(18,9,134)
2430 CALL HCHAR(18,7,128)
2440 CALL HCHAR(18,5,134)
2450 CALL HCHAR(18,8,140)
2460 GOTO 2230
2470 REM *** AL PASSAGGIO ***
2480 IF M=1 THEN 2540
2490 IF (R2+C2)/2=INT((R2+C2)/2) THEN 2520
2500 CALL HCHAR(15,C2,158)
2510 GOTO 2230
2520 CALL HCHAR(15,C2,134)
2530 GOTO 2230
2540 IF (R2+C2)/2=INT((R2+C2)/2) THEN 2570
2550 CALL HCHAR(14,C2,158)
2560 GOTO 2230
2570 CALL HCHAR(14,C2,134)
2580 GOTO 2230
2590 REM *** PROMOZIONE ***
2600 ORD=(R2+C2)/2
2610 IF ORD=INT(ORD) THEN 2670
2620 IF M=1 THEN 2630
2630 PEZZO=YY(PRO)+16
2640 GOTO 2710
2650 PEZZO=YY(PRO)
2660 GOTO 2710
2670 IF M=1 THEN 2700
2680 PEZZO=YY(PRO)+24
2690 GOTO 2710
2700 PEZZO=YY(PRO)+8
2710 CALL HCHAR(R2,C2,PEZZO)
2720 GOTO 2230
2730 REM *** FINE PARTITA ***
2740 CALL HCHAR(23,1,32,64)
2750 PRINT "VUOI RIVEDERE LA STESSA"
2760 INPUT "PARTITA? (S/N) " : RIS$
2770 IF SEG$(RIS$,1,1)="S" THEN 1330
2780 GOTO 190
2790 CALL CLEAR
2800 END

```

Alla linea 190 viene dimensionata la matrice unidimensionale A#, ogni suo elemento contiene una mossa. Con l'istruzione 360 si inizia ad introdurre i dati relativi ad una partita; alcuni messaggi guidano l'utente in questa fase. L'istruzione 660 stampa i numeri progressivi; la funzione TAB fa in modo che risultino incolonnati a destra. Le stringhe che definiscono le mosse devono risultare necessariamente lunghe 12 caratteri altrimenti la linea 590 rileva un errore; se la stringa termina con un punto l'istruzione 620 fa passare alla fase di registrazione.

Il file viene aperto alla stringa 680 ed ogni record contiene un blocco di 4 stringhe.

Alla riga 770 inizia la sezione per il richiamo e l'esecuzione di una partita. Tutti i caratteri grafici definiti alle linee 780-1170 servono a disegnare la scacchiera ed i pezzi; ricordiamo che lo stesso pezzo su casa bianca e su casa nera deve essere definito con due diversi caratteri. Le variabili con indice YY alle righe 1180-1210 serviranno in seguito per la promozione di un pedone. L'istruzione OPEN apre il file per la lettura dei dati precedentemente registrati; quando appare la scritta REWIND CASSETTE TAPE bisogna posizionare il nastro esattamente dove si sa essere registrata proprio la partita che si vuole rivedere. Dopo che la partita è ritornata nella memoria centrale ogni record viene scisso nelle quattro stringhe ed ogni stringa nelle due semistringhe (istruzione 1900). La prima cosa che viene valutata è se la mossa è un arrocco o se la semistringa contiene il punto finale che indica il termine della partita (linee 1910-1930). Le istruzioni 1940-1970 decodificano le coordinate di ogni mossa trasformandole in numeri di riga e di colonna.

R1 e C1 si riferiscono alla casa di partenza mentre R2 e C2 a quella di arrivo. L'istruzione CALL GCHAR 1980 riconosce il carattere posto in R1 e C1 e pone il suo numero di codice in PX. Il numero di carattere resterà invariato dopo lo spostamento se resterà invariato il colore della casella ma dovrà variare al variare di quello.

Il computer sa se una casa è chiara o scura sommando il numero di riga con il numero di colonna, se la somma è pari la casa è chiara, se è dispari è scura.

Da 2140 a 2220 c'è il controllo di una eventuale presa al passaggio o sulla promozione di un pedone per le quali si rimanda alle rispettive routine che occupano l'ultima parte del listato.

È stata introdotta la variabile M per far ripetere il ciclo due volte, una per la semistringa di sinistra (mossa del bianco) ed una per quella di destra (mossa del nero); M infatti rappresenta il punto dal quale la funzione SEG\$ di linea 1900 inizia la lettura della stringa.

Memory

È la versione computerizzata di un gioco conosciuto da molti che mette alla prova le capacità di memoria visiva.

Un certo numero di tessere, nel nostro caso 20, portano impresse su una faccia dei disegni che si ripetono a formare delle coppie. Le tessere vengono mescolate e disposte capovolte; i due giocatori a turno scoprono due carte, se portano impresso lo stesso disegno chi è di mano ha realizzato un punto ed il gioco resta allo stesso concorrente, altrimenti le tessere sono nuovamente capovolte ed il gioco passa all'avversario. Vince

chi, ricordando la posizione dei vari disegni, riesce a scoprire più coppie. Nel programma i disegni sono realizzati con degli sprite che utilizzano caratteri definiti su una matrice di 16x16 pixel e successivamente ingranditi con la funzione CALL MAGNIFY. In TI Basic teoricamente si sarebbero dovuti definire 160 caratteri grafici per ottenere lo stesso effetto.

Le tessere sono individuate in base alle coordinate; la prima, una lettera dalla A alla E, indica la colonna, la seconda, un numero da 1 a 4, indica la riga. Il programma è protetto da eventuali errori.

```

000 REM *****
010 REM *
020 REM * MEMORY *
030 REM *
040 REM *****
050 REM di Sergio Borsani
060 REM versione:
070 REM TI EXTENDED BASIC
080 CALL CLEAR
090 CALL COLOR(2,16,7)
100 CALL HCHAR(7,1,42,32)
110 CALL HCHAR(10,1,42,3)
120 CALL HCHAR(11,2,42,5)
130 CALL VCHAR(10,5,42,5)
140 CALL VCHAR(10,7,42,3)
150 CALL HCHAR(14,8,42,3)
160 CALL HCHAR(11,13,42)
170 CALL VCHAR(10,16,42,5)
180 CALL HCHAR(10,18,42,3)
190 CALL VCHAR(10,21,42,5)
200 CALL HCHAR(10,24,42,3)
210 CALL HCHAR(13,25,42)
220 CALL HCHAR(10,28,42)
230 CALL HCHAR(10,32,42)
240 CALL HCHAR(11,3,42)
250 CALL HCHAR(11,13,42)
260 CALL HCHAR(11,29,42)
270 CALL HCHAR(11,31,42)
280 CALL HCHAR(11,29,42)
290 CALL HCHAR(11,29,42)
300 CALL HCHAR(11,29,42)
310 CALL HCHAR(11,29,42)
320 CALL HCHAR(11,29,42)
330 CALL HCHAR(11,29,42)
340 CALL HCHAR(11,29,42)
350 CALL HCHAR(11,29,42)
360 CALL HCHAR(11,29,42)
370 CALL HCHAR(11,29,42)
380 CALL HCHAR(11,29,42)
390 CALL HCHAR(11,29,42)
400 CALL HCHAR(11,29,42)
410 CALL HCHAR(11,29,42)
420 CALL HCHAR(11,29,42)
430 CALL HCHAR(11,29,42)
440 CALL HCHAR(11,29,42)
450 CALL HCHAR(11,29,42)
460 CALL HCHAR(11,29,42)
470 CALL HCHAR(11,29,42)
480 CALL HCHAR(11,29,42)
490 CALL HCHAR(11,29,42)
500 CALL HCHAR(11,29,42)
510 CALL HCHAR(11,29,42)
520 CALL HCHAR(11,29,42)
530 CALL HCHAR(11,29,42)
540 CALL HCHAR(11,29,42)
550 CALL HCHAR(11,29,42)
560 CALL HCHAR(11,29,42)
570 CALL HCHAR(11,29,42)
580 CALL HCHAR(11,29,42)
590 CALL HCHAR(11,29,42)
600 CALL HCHAR(11,29,42)
610 CALL HCHAR(11,29,42)
620 CALL HCHAR(11,29,42)
630 CALL HCHAR(11,29,42)
640 CALL HCHAR(11,29,42)
650 CALL HCHAR(11,29,42)
660 CALL HCHAR(11,29,42)
670 CALL HCHAR(11,29,42)
680 CALL HCHAR(11,29,42)
690 CALL HCHAR(11,29,42)
700 CALL HCHAR(11,29,42)
710 CALL HCHAR(11,29,42)
720 CALL HCHAR(11,29,42)
730 CALL HCHAR(11,29,42)
740 CALL HCHAR(11,29,42)
750 CALL HCHAR(11,29,42)
760 CALL HCHAR(11,29,42)
770 CALL HCHAR(11,29,42)
780 CALL HCHAR(11,29,42)
790 CALL HCHAR(11,29,42)
800 CALL HCHAR(11,29,42)
810 CALL HCHAR(11,29,42)
820 CALL HCHAR(11,29,42)
830 CALL HCHAR(11,29,42)
840 CALL HCHAR(11,29,42)
850 CALL HCHAR(11,29,42)
860 CALL HCHAR(11,29,42)
870 CALL HCHAR(11,29,42)
880 CALL HCHAR(11,29,42)
890 CALL HCHAR(11,29,42)
900 CALL HCHAR(11,29,42)
910 CALL HCHAR(11,29,42)
920 CALL HCHAR(11,29,42)
930 CALL HCHAR(11,29,42)
940 CALL HCHAR(11,29,42)
950 CALL HCHAR(11,29,42)
960 CALL HCHAR(11,29,42)
970 CALL HCHAR(11,29,42)
980 CALL HCHAR(11,29,42)
990 CALL HCHAR(11,29,42)

```

```

770 NEXT RIGA
780 FOR L=1 TO 5
790 CALL HCHAR(1,1+L*3,44+L)
800 IF L=5 THEN 820
810 CALL HCHAR(-1+L*3,3,48+L)
820 NEXT L
830 CARATT=104
840 RANDOMIZE
850 JF=INT(RND*20)+1
860 IF C(J)<>0 THEN 850
870 C(J)=CARATT : CARATT=CARATT+4
880 IF CARATT<141 THEN 850
890 CARATT=104
900 K=INT(RND*20)+1
910 IF C(K)<>0 THEN 940
920 C(K)=CARATT : CARATT=CARATT+4
930 IF CARATT<141 THEN 900 ELSE 960
940 K=K+1
950 IF K=20 THEN 910 ELSE K=1 : GOTO 910
960 DISPLAY AT(23,3):"PUNTI"
970 DISPLAY AT(23,22):"PUNTI"
980 P1=0 : P2=0 : B="A"
990 DISPLAY AT(24,4)BEEP SIZE(4):P1
1000 DISPLAY AT(24,23)BEEP:P2
1010 IF P1+P2=10 THEN 1610
1020 DISPLAY AT(23,11)SIZE(6):"GIOCA!"
1030 DISPLAY AT(23,18)SIZE(1):B
1040 ACCEPT AT(24,14)BEEP VALIDATE("ABCDE1234")SIZE(2):XY
1050 IF LEN(XY)=2 THEN 1100
1060 DISPLAY AT(24,11)SIZE(12)BEEP:"ERRORE !"
1070 FOR T=1 TO 500 : NEXT T
1080 DISPLAY AT(24,9)SIZE(12):" "
1090 GOTO 1040
1100 IF ASC(SEG$(XY$,1,1))<65 THEN 1060
1110 IF ASC(SEG$(XY$,2,1))>52 THEN 1060
1120 N1=ASC(SEG$(XY$,1,1))-64
1130 IF N1<1 OR N1>5 THEN 1060
1140 N2=VAL(SEG$(XY$,2,1))
1150 IF N2<1 OR N2>4 THEN 1060
1160 NN=N1+(N2-1)*5
1170 IF B(NN)=1 THEN 1060
1180 IF C(NN)=104 OR C(NN)=140 THEN COLORE=6 : GOTO 1240
1190 IF C(NN)=108 OR C(NN)=116 THEN COLORE=2 : GOTO 1240
1200 IF C(NN)=112 OR C(NN)=120 THEN COLORE=9 : GOTO 1240
1210 IF C(NN)=124 THEN COLORE=11 : GOTO 1240
1220 IF C(NN)=128 OR C(NN)=132 THEN COLORE=5 : GOTO 1240
1230 IF C(NN)=136 THEN COLORE=13
1240 IF SCELTA1 THEN 1260
1250 SCELTA1=C(NN): PRIMO=NN : GOTO 1270
1260 IF NN=PRIMO THEN 1060 : SCELTA2=C(NN)
1270 FOR RIBA=N2*5-2 TO N2*5+1 : CALL HCHAR(RIBA,N1*3,97,4): NEXT RIBA
1280 CALL SPRITE(NN,C(NN),COLORE,N2*40-23,N1*40-7)
1290 CALL MAGNIFY(4)
1300 IF SCELTA2=0 THEN 1040
1310 FOR T=1 TO 500 : NEXT T
1320 IF SCELTA1<>SCELTA2 THEN 1470
1330 B(PRIMO)=1 : B(NN)=1
1340 PRIMO=0 : SCELTA1=0 : SCELTA2=0
1350 IF B="B" THEN 1370
1360 P1=P1+1 : GOTO 1380
1370 P2=P2+1
1380 CALL SOUND(150,440,2)
1390 CALL SOUND(150,466,2)
1400 CALL SOUND(300,523,2)
1410 CALL SOUND(300,440,2)
1420 CALL SOUND(300,587,2)
1430 CALL SOUND(300,784,2)
1440 CALL SOUND(300,659,2)
1450 FOR T=1 TO 500 : NEXT T
1460 CALL DELSPRITE(ALL): GOTO 990
1470 VOLTE=1
1480 IF VOLTE=1 THEN 1510
1490 CALL POSITION(NN,X1,Y1): VOLTE=2
1500 CALL DELSPRITE(NN): GOTO 1530
1510 CALL POSITION(NPRIMO,X1,Y1)
1520 CALL DELSPRITE(NPRIMO)
1530 RIBA=INT((X1+2)/8)+1 : COLD=INT(Y1/8)+1
1540 FOR I=0 TO 3
1550 CALL HCHAR(RIBA+I,COLD,96,4)
1560 NEXT I
1570 IF VOLTE=1 THEN 1490
1580 PRIMO=0 : SCELTA1=0 : SCELTA2=0
1590 IF B="A" THEN B="B" : GOTO 1030
1600 B="A" : GOTO 1030
1610 FOR T=1 TO 1000 : NEXT T
1620 CALL DELSPRITE(ALL)

```

```

1630 CALL SOUND(300,440,2):: CALL SOUND(150,466,2)
1640 CALL SOUND(150,523,2):: CALL SOUND(150,587,2)
1650 CALL SOUND(150,523,2):: CALL SOUND(300,466,2):: CALL SOUND(300,440,2)
1660 CALL SOUND(1000,392,2,330,5)
1670 FOR T=1 TO 300 :: NEXT T
1680 CALL CLEAR :: CALL COLOR(2,2,1)
1690 DISPLAY AT(12,3):"VUDI RICOMINCIARE? (S/N) "
1700 CALL KEY(0,K,S)
1710 IF K=78 OR K=110 THEN 1760
1720 IF K<>83 AND K<>115 THEN 1700
1730 PRIMO=0 :: SCELTA1=0 :: SCELTA2=0
1740 FOR I=1 TO 20 :: B(I)=0 :: C(I)=0 :: NEXT I
1750 CALL CLEAR :: GOTO 700
1760 DISPLAY AT(12,13)ERASE ALL:"FINE"
1770 FOR T=1 TO 500 :: NEXT T
1780 CALL CLEAR
1790 END

```

Le istruzioni 200-340 scrivono il titolo in formato gigante utilizzando il carattere 42, cioè l'asterisco. Con un ciclo FOR-NEXT il colore passa da bianco su fondo rosso a rosso su fondo bianco simulando delle luci lampeggianti.

Alla riga 530, A# contiene i disegni o meglio le stringhe esadecimali che li definiscono, B memorizza la posizione delle tessere che non devono più essere chiamate perché contengono le coppie già scoperte, C invece contiene i numeri dei caratteri che costituiscono i disegni nell'ordine stabilito dalla funzione RND. Alle righe 570-660 si trovano le stringhe che definiscono i disegni; essi sono nell'ordine: pesce, Charlot, cuore, ombrello, la lettera M, casa, scacchiera, barca, albero e automobile.

I caratteri sono numerati da 104 a 140.

Le istruzioni 700-770 disegnano le tessere capovolte di colore rosso mentre le righe successive scrivono le coordinate su due lati dello schermo. Da riga 830 a riga 960 le dieci coppie di caratteri vengono disposte casualmente all'interno del vettore C; le locazioni ove appaiono i disegni sono numerate sequenzialmente procedendo da sinistra a destra e dall'alto in basso. Terminata questa operazione il computer è pronto per accogliere le chiamate dei concorrenti.

Abbiamo già detto che il programma è protetto da eventuali errori; il primo accorgimento è l'opzione VALIDATE, nell'ACCEPT di riga 1040, che accetta solo i caratteri alfanumerici specificati, poi si verifica che le coordinate contengano effettivamente due caratteri e che la lettera preceda il numero.

La variabile NN (riga 1160) è un numero da 1 a 20 che si riferisce alla casella chiamata; l'istruzione successiva controlla che la tessera non sia già scoperta, solo allora si passa alla stampa del disegno.

Ogni giocatore deve scoprire due tessere, SCELTA1 e SCELTA2 contengono i numeri dei caratteri associati alle caselle chiamate. Solo se SCELTA1=SCELTA2 si incrementa il punteggio del giocatore (righe 1360-1370), si pongono uguali ad uno gli elementi del vettore B corrispondenti alle caselle per impedire che le stesse vengano chiamate un'altra volta e si toglie lo sprite prima di ricominciare. Sarebbe stato forse più piacevole mantenere i disegni sullo schermo, tuttavia in TI Extended

Basic non possono coesistere più di 4 sprite sulla stessa riga e l'ultimo di questi sarebbe comunque scomparso.

L'ultimo perde

È il gioco dei 16 bastoncini; si tolgono a turno quanti bastoncini si desidera, ma solo da una delle quattro file. Perde chi resta con l'ultimo. Come si vede questa condizione ha dato il nome al programma; il gioco ha certamente una sua denominazione ma francamente non la conosciamo con sicurezza.

Quest'ultimo programma vuole dimostrare come in simili applicazioni il computer possa non solo controllare lo svolgimento del gioco ma avere una sua strategia vincente.

È la filosofia che ha portato alla costruzione dei computer che giocano a scacchi, a dama o a backgammon. Alla base di tutto sta la straordinaria velocità di calcolo del computer. Un buon giocatore di scacchi si affida spesso alla intuizione ed ogni volta controlla forse una decina di sequenze di mosse che comprendono le proprie e quelle probabili dell'avversario. Il computer, che non possiede alcunché di paragonabile all'intuizione, può tuttavia controllare migliaia di varianti e valutare per ognuna di esse la situazione che si è venuta a creare al termine.



Lo studio delle varianti procede generalmente per una profondità di due o tre mosse, dopo di che il loro numero diventa così elevato da portare il tempo di elaborazione ad un limite non più accettabile.

In questo programma il computer considera tutte le varianti con una profondità di una sola mossa, cioè esso valuta tutte le proprie mosse e

per ciascuna tutte le possibili risposte dell'avversario. In questo modo l'elaborazione risulta veloce ed il livello di gioco abbastanza buono. Poiché chi inizia, in questo caso, è svantaggiato, il computer estrae a sorte a chi spetta il tratto; la mossa si effettua specificando la lettera che contraddistingue ogni fila e il numero di bastoncini che si desidera togliere. Se è il computer ad iniziare, la sua prima mossa è casuale, infatti non ce n'è una che possa dirsi migliore di un'altra. In questo gioco le mosse o sono vincenti o sono perdenti, non esiste una graduazione e non è necessario attribuire un punteggio dopo ogni possibile risposta dell'avversario. Quando non sia la prima, il computer esamina ogni mossa possibile per individuarne una vincente; se non la trova, ne esegue virtualmente una vincente; se non la trova, ne esegue virtualmente una e per essa considera le possibili risposte dell'avversario. Se nessuna di queste porta l'avversario ad una situazione vincente allora la esegue altrimenti la cambia e ritorna a studiare le risposte.

Può succedere che nessuna mossa consenta al computer di evitare la sconfitta ed in questo caso esegue semplicemente l'ultima presa in considerazione "sperando" in un errore da parte dell'avversario.

Per giudicare se una mossa è vincente basta esaminare i bastoncini rimasti. Se rimane un solo bastoncino all'avversario, egli ha perso; ma perderà certamente anche se rimane con tre bastoncini su tre file diverse. Il lettore può verificare per tentativi che si perde anche con due bastoncini su una fila e due in un'altra o se si rimane con un bastoncino in una fila, due in un'altra e tre in un'altra ancora.

Vediamo ora il listato.

```

100 REM      L'ULTIMO PERDE
110 REM      *****
120 REM      di: Sergio Borsani
130 REM      versione: TI BASIC
140 CALL CLEAR
150 CALL CHAR(128,"F")
160 CALL COLOR(13,5,5)
170 PRINT TAB(8);"ULTIMO PERDE"::::::::::::
180 CALL MCHAR(10,8,128,18)
190 CALL VCHAR(11,8,128,3)
200 CALL VCHAR(11,25,128,3)
210 CALL MCHAR(14,8,128,18)
220 CALL SOUND(400,440,8)
230 CALL SOUND(200,392,8)
240 CALL SOUND(200,440,7)
250 CALL SOUND(400,587,7)
260 CALL SOUND(200,523,5)
270 CALL SOUND(200,587,4)
280 CALL SOUND(400,494,3)
290 CALL SOUND(200,440,4)
300 CALL SOUND(200,494,5)
310 CALL SOUND(500,392,6)
320 CALL CHAR(136,"FFFFFFFFFFFFFF")
330 CALL CHAR(137,"00C0F0FBFCFCFE")
340 CALL CHAR(138,"FEFECFCFCFC")
350 CALL CHAR(139,"7F7F3F3F1F0F03")
360 CALL CHAR(140,"00030F1F3F3F7F7F")
370 CALL CHAR(144,"003B4440404043B")
380 CALL CHAR(145,"007C101010101010")
390 CALL COLOR(14,12,1)
400 CALL COLOR(15,7,12)
410 FFS="SCRIVI LA TUA MOSSA"
420 FRS="LA ESEGUI? (S/N)"
430 FTS="LA MIA MOSSA E:"
440 REM      *** INIZIALIZZAZIONE ***

```

```

450 R(1)=1
460 R(2)=3
470 R(3)=5
480 R(4)=7
490 CALL CLEAR
500 PRINT "PREMI:";
510 PRINT "1. PER LE SPIEGAZIONI";
520 PRINT "2. PER GIOCARRE";
530 CALL KEY(0,K,S)
540 IF S=0 THEN 530
550 IF K=49 THEN 590
560 IF K=50 THEN 730
570 GOTO 530
580 REM *** SPIEGAZIONI ***
590 CALL CLEAR
600 PRINT "SULLO SCHERMO APPARIRANNO 16"
610 PRINT "BASTONCINI DISPOSTI SU 4 FI-"
620 PRINT "LE DENOMINATE A,B,C,D.";
630 PRINT "IL GIOCO CONSISTE NEL TO-"
640 PRINT "GLIERE A TURNO QUANTI BA-"
650 PRINT "STONCINI SI VUOLE DA UNA FI-"; "LA."
660 PRINT "PERDE CHI RESTA CON L'ULTIMO";
670 PRINT "PER MUOVERE SPECIFICARE LA"
680 PRINT "LETTERA CORRISPONDENTE ALLA"
690 PRINT "RIGA E IL NUMERO DI BASTON-"; "CINI."
700 PRINT "IL SORTEGGIO STABILISCE CHI DEVE GIOCARRE.";
710 GOSUB 2210
720 GOTO 490
730 CALL CLEAR
740 PRINT "SORTEGGIAMO CHI DEVE INIZIA-"; "RE.";
750 PRINT "SCEGLI TESTA O CROCE? (T/C)";
760 CALL KEY(0,KK,S)
770 IF S=0 THEN 780
780 IF KK=84 THEN 810
790 IF KK=67 THEN 830
800 GOTO 740
810 PRINT "BENE, HAI SCELTO TESTA"
820 GOTO 840
830 PRINT "BENE, HAI SCELTO CROCE"
840 PRINT "11111111111111111111"
850 RANDOMIZE
860 SORTE=INT(RND*10)+1
870 CALL HCHAR(18,15,140)
880 CALL HCHAR(18,16,136)
890 CALL HCHAR(18,17,137)
900 CALL HCHAR(19,15,136,3)
910 CALL HCHAR(20,15,139)
920 CALL HCHAR(20,16,136)
930 CALL HCHAR(20,17,138)
940 FOR I=1 TO 10
950 CALL HCHAR(19,16,145)
960 CALL HCHAR(19,16,144)
970 NEXT I
980 IF SORTE>5 THEN 1020
990 SORTE=84
1000 CALL HCHAR(19,16,145)
1010 GOTO 1040
1020 SORTE=67
1030 CALL HCHAR(19,16,144)
1040 IF KK=SORTE THEN 1070
1050 PRINT "HAI PERSO, COMINCIO IO.";
1060 GOTO 1080
1070 PRINT "HAI VINTO, COMINCI TU.";
1080 GOSUB 2210
1090 CALL CLEAR
1100 DATA 16,16,14,18,12,20,10,22
1110 RESTORE 1100
1120 FOR I=1 TO 4
1130 READ PRIMO,ULTIMO
1140 FOR J=PRIMO TO ULTIMO STEP 2
1150 CALL HCHAR(I*4+1,6,64+I)
1160 CALL VCHAR(I*4,J,128,3)
1170 NEXT J
1180 NEXT I
1190 CALL HCHAR(20,3,136,28)
1200 IF KK=SORTE THEN 1230 ELSE 2260
1210 REM *** INIZIA IL ***
1220 REM *** CONCORRENTE ***
1230 IF R(1)+R(2)+R(3)+R(4)<>1 THEN 1270
1240 PRINT "MI DISPIACE, HAI PERSO!";
1250 GOSUB 2730
1260 GOTO 2840
1270 FOR J=1 TO LEN(FF$)
1280 L=ASC(SEG$(FF$,J,1))
1290 CALL HCHAR(22,2+J,L)
1300 NEXT J

```

```

1310 CALL KEY(0,K1,S)
1320 IF S=0 THEN 1310
1330 IF (K1<45)+(K1>48) THEN 1310
1340 IF R(K1-64)=0 THEN 1310
1350 CALL HCHAR(22,24,K1)
1360 CALL KEY(0,K2,S)
1370 IF S=0 THEN 1360
1380 IF K2<49 THEN 1360
1390 IF K2=48+R(K1-64) THEN 1360
1400 CALL HCHAR(22,25,K2)
1410 FOR J=1 TO LEN(FA*)
1420 L=ASC(SEG$(FA*,J,1))
1430 CALL HCHAR(24,2+J,L)
1440 NEXT J
1450 CALL KEY(0,K,S)
1460 IF S=0 THEN 1450
1470 IF K=83 THEN 1520
1480 IF K=78 THEN 1500
1490 GOTO 1450
1500 CALL HCHAR(22,24,32,72)
1510 GOTO 1310
1520 RIGA=K1-64
1530 NUMERO=K2-48
1540 GOSUB 2330
1550 CALL HCHAR(22,1,32,96)
1560 REM *** GIOCA TI-99 ***
1570 IF R(1)+R(2)+R(3)+R(4)<>1 THEN 1610
1580 PRINT "HA1 VINTO!":
1590 GOSUB 2430
1600 GOTO 2840
1610 FOR I=1 TO 4
1620 RR(I)=R(I)
1630 NEXT I
1640 RIGA=0
1650 NUMERO=0
1660 RIGA=RIGA+1
1670 IF RIGA>4 THEN 1800
1680 NUMERO=NUMERO+1
1690 IF NUMERO>RR(RIGA) THEN 1650
1700 MOSSA1=RIGA
1710 MOSSA2=NUMERO
1720 RR(RIGA)=RR(RIGA)-NUMERO
1730 IF RR(1)+RR(2)+RR(3)+RR(4)=1 THEN 2100
1740 GOSUB 2390
1750 IF PERDE=1 THEN 2100
1760 RR(RIGA)=RR(RIGA)+NUMERO
1770 GOTO 1680
1780 REM ***PROBABILI MOSSE***
1790 REM ***DELL'AVVERSARIO***
1800 RIGA=4
1810 GOTO 1970
1820 RA=0
1830 NA=0
1840 RA=RA+1
1850 IF RA>4 THEN 2080
1860 NA=NA+1
1870 IF NA>RR(RA) THEN 1830
1880 RR(RA)=RR(RA)-NA
1890 IF RR(1)+RR(2)+RR(3)+RR(4)=1 THEN 1920
1900 GOSUB 2390
1910 IF PERDE=0 THEN 2050
1920 RR(RIGA)=RR(RIGA)+NUMERO
1930 RR(RA)=RR(RA)+NA
1940 GOTO 1970
1950 RIGA=RIGA-1
1960 IF RIGA=0 THEN 2080
1970 NUMERO=NUMERO+1
1980 IF NUMERO<RR(RIGA) THEN 2010
1990 NUMERO=0
2000 GOTO 1930
2010 MOSSA1=RIGA
2020 MOSSA2=NUMERO
2030 RR(RIGA)=RR(RIGA)-NUMERO
2040 GOTO 1820
2050 RR(RA)=RR(RA)+NA
2060 GOTO 1860
2070 REM *** MOSSA TI-99 ***
2080 RIGA=MOSSA1
2090 NUMERO=MOSSA2
2100 FOR J=1 TO LEN(FT*)
2110 L=ASC(SEG$(FT*,J,1))
2120 CALL HCHAR(22,2+J,L)
2130 NEXT J

```

```

2140 CALL HCHAR(22,24,64+RIGA)
2150 CALL HCHAR(22,25,48+NUMERO)
2160 FOR T=1 TO 300
2170 NEXT T
2180 GOSUB 2330
2190 CALL HCHAR(22,3,32,28)
2200 GOTO 1230
2210 PRINT TAB(6);"- PREMI UN TASTO -"
2220 CALL KEY(0,K,B)
2230 IF B=0 THEN 2220
2240 RETURN
2250 REM *** PRIMA MOSSA ***
2260 RANDOMIZE
2270 RIGA=INT(RND*4)+1
2280 NUMERO=1
2290 FOR T=1 TO 300
2300 NEXT T
2310 GOTO 2100
2320 REM *** CANCELLA ***
2330 CC=(4-RIGA)*2+10+2*R(RIGA)
2340 R(RIGA)=R(RIGA)-NUMERO
2350 FOR I=1 TO NUMERO
2360 CALL VCHAR(RIGA*4,CC-I*2,32,3)
2370 NEXT I
2380 RETURN
2390 FOR I=1 TO 4
2400 RW(I)=RR(I)
2410 NEXT I
2420 FOR B=1 TO 3
2430 F=0
2440 FOR C=1 TO 4-B
2450 IF RW(C+1)>RW(C) THEN 2500
2460 T=RW(C)
2470 RW(C)=RW(C+1)
2480 RW(C+1)=T
2490 F=F+1
2500 NEXT C
2510 IF F=0 THEN 2530
2520 NEXT B
2530 IF (RW(1)=0)*(RW(2)=1)*(RW(3)=1)*(RW(4)=1) THEN 2610
2540 IF (RW(1)=0)*(RW(2)=1)*(RW(3)=4)*(RW(4)=5) THEN 2610
2550 IF (RW(1)=0)*(RW(2)=1)*(RW(3)=2)*(RW(4)=3) THEN 2610
2560 IF (RW(1)=0)*(RW(2)=2)*(RW(3)=4)*(RW(4)=6) THEN 2610
2570 IF (RW(1)=RW(2))*(RW(3)=RW(4)) THEN 2600
2580 PERDE=0
2590 GOTO 2620
2600 IF RW(3)=1 THEN 2580
2610 PERDE=1
2620 RETURN
2630 CALL SOUND(150,659,2)
2640 CALL SOUND(150,587,2)
2650 CALL SOUND(300,523,2)
2660 CALL SOUND(300,440,2)
2670 CALL SOUND(300,698,2,440,5)
2680 CALL SOUND(300,784,2,587,5)
2690 CALL SOUND(150,698,2,392,5)
2700 CALL SOUND(150,659,2)
2710 CALL SOUND(1000,698,2,440,5)
2720 RETURN
2730 CALL SOUND(300,523,2)
2740 CALL SOUND(300,440,2)
2750 CALL SOUND(150,440,2)
2760 CALL SOUND(150,494,2)
2770 CALL SOUND(300,523,2)
2780 CALL SOUND(150,494,2)
2790 CALL SOUND(150,523,2)
2800 CALL SOUND(150,494,2)
2810 CALL SOUND(150,392,2)
2820 CALL SOUND(1000,440,2)
2830 RETURN
2840 PRINT "VUOI RICOMINCIARE? (S/N) ";
2850 CALL KEY(0,K,B)
2860 IF B=0 THEN 2850
2870 IF K=83 THEN 440
2880 IF K=78 THEN 2910
2890 PRINT "ERRORE! RIPROVA";
2900 GOTO 2840
2910 CALL CLEAR
2920 END

```

Il carattere 128 (istruzione 150) è utilizzato per creare una cornice attorno al titolo e soprattutto per disegnare i bastoncini indispensabili per il gioco. Gli altri caratteri (linee 320-380) consentono di disegnare una mo-

neta per sorteggiare a testa o croce chi deve iniziare. Le stringhe FF\$, FR\$ e FT\$ sono dei messaggi che compariranno nel corso del gioco. Il vettore R (righe 450-480) contiene il numero dei bastoncini presenti in ogni momento in ogni riga.

Dopo una parte dedicata alle spiegazioni delle regole del gioco si passa alla parte attiva con il sorteggio della prima mossa (linea 740-1080); se questa spetta al concorrente si passa all'istruzione 1230, se invece spetta al computer si salta alla linea 2260. Questa parte del programma viene utilizzata dal computer solo per eseguire la sua prima mossa che, come è stato detto, è casuale; per esser più precisi è casuale solo la scelta della riga perché viene tolto sempre un solo bastoncino, come detta la prudenza!

Poiché il programma ritorna ciclicamente sui suoi passi, prima di eseguire qualsiasi mossa si controlla sempre che non sia rimasto un solo bastoncino; la qual cosa si verifica sommando le quattro variabili con indice R (istruzione 1230). I cicli FOR-NEXT alle linee 1270-1300 e 1410-1440 stampano i messaggi visti in precedenza evitando lo scrolling del video verso l'alto, simulando l'istruzione DISPLAY AT disponibile in Extended Basic. In modo analogo le istruzioni CALL KEY alle righe 1310 e 1360 sostituiscono l'istruzione INPUT. K1 e K2 contengono i codici ASCII dei due tasti premuti per definire una mossa, esse vengono poi trasformate nelle variabili RIGA e NUMERO che si riferiscono la prima alla fila dalla quale si tolgono i bastoncini e la seconda al numero dei bastoncini stessi. La routine 2320 effettua la mossa togliendo i bastoncini e aggiornando il vettore R.

Per cercare una mossa che gli consenta di vincere il computer trasferisce per prima cosa i valori di R in RR (1610-1630) al fine di non alterare, nel corso dell'elaborazione, il contenuto del primo vettore; dopo di che prova a togliere un bastoncino dalla quarta riga, poi due, poi tre, fino a quanti ne contiene la riga stessa; procede quindi togliendo un bastoncino dalla terza riga, e così via. Per non sommare le conseguenze di due mosse, prima di ognuna il computer ripristina l'originale contenuto di RR. Ad ogni tentativo la routine 2390 controlla se si è venuta a creare una situazione vincente. Vediamo subito come lavora questa routine. Abbiamo detto che se rimangono, ad esempio, tre file rispettivamente con 1, 2 e 3 bastoncini l'avversario non può salvarsi e la situazione è da considerarsi vincente.

In questo caso il contenuto delle variabili con indice RR sarà 0, 1, 2, e 3; tuttavia potrebbe essere anche 1, 0, 3, 2 o qualsiasi altra disposizione che non contenga 2 o 3 al primo posto. Lo stesso dicasi per le altre situazioni vincenti. Per non dover considerare un numero troppo elevato di casi si trasferisce il vettore RR in RW e questo si ordina secondo valori crescenti; in seguito si dovrà controllare la presenza di una sola disposizione per ogni situazione vincente. Al termine della routine, la variabi-

le flag PERDE segnala l'esito del controllo; se PERDE = 1 il computer abbandona le ricerche ed esegue senz'altro la mossa che era stata memorizzata in MOSSA1 (numero della fila) e MOSSA2 (numero dei bastoncini).

Se non viene trovata una mossa vincente il programma passa alla linea 1800 per esaminare le possibili mosse dell'avversario; verrà eseguita quella mossa, se esiste, che non consente risposte vincenti da parte dell'avversario.

Anche in questa parte MOSSA1 e MOSSA2 hanno lo stesso significato dato loro in precedenza, RA ed NA rappresentano rispettivamente la riga ed il numero di bastoncini giocabili dall'avversario e corrispondono pertanto ad una probabile mossa. Quando si verifica la condizione di riga 1850, sono state esaminate tutte le possibili mosse dell'avversario senza che la routine 2390 abbia rilevato una situazione vincente; la mossa memorizzata viene allora effettivamente giocata.

Una caratteristica che a qualcuno potrà sembrare un difetto è che, se l'avversario inizia sempre con una mossa, anche il programma elabora sempre la medesima risposta creando sequenze che possono facilmente ripetersi in un'altra partita; tuttavia ciò è dovuto in gran parte alla conformazione del gioco.

È interessante notare che simili tecniche di programmazione sono applicabili non solo ad una vasta gamma di giochi ove sia necessario dotare il computer di una strategia vincente ma anche nell'ambito di problemi di programmazione lineare in cui l'obiettivo sia la ricerca di situazioni ottimali come ad esempio il migliore sfruttamento delle risorse, la riduzione dei costi, ecc.

Indice analitico

- algoritmo euclideo 16
- angoli 48
- angolo parallattico 50
- archivio scacchistico 190
- area di una superficie limitata da una curva 115
- calcolo combinatorio 117
- cinematica 159
- città d'Italia 187
- combinazioni semplici 126
- composizione dei moti 168
- cono 37
- Cramer (regola di) 102
- curva a campana 134
- derivata 87
- determinanti 102
- disposizioni semplici 117
- distribuzione binomiale 142
- distribuzione normale di probabilità 132
- eccentricità (di un'orbita) 175
- ellisse 175
- equazione di una curva passante per dati punti 109
- equilibrio di un corpo rigido 150
- equilibrio di una trave 158
- forze 148
- frazioni 19
- funzione 87
- Gauss (metodo di) 111
- gaussiana 132
- geometria analitica 71-77
- gittata 169
- grafica 57
- grafica a media risoluzione 59
- Horner-Ruffini (algoritmo di) 97
- integrali definiti 110
- interesse composto 27
- interesse semplice 26
- Keplero (leggi di) 174
- lunghezza di un arco di parabola 115
- massimi e minimi relativi 88
- massimo comune divisore 17
- media aritmetica 132
- memory 196
- metodo di Newton 96
- minimo comune multiplo 17

- momento di una forza 150
moto ad accelerazione costante 168
moto rettilineo uniforme 159
moto uniformemente accelerato 159
orbite 174
orbite circolari 177
ortogrammi 64
parabole 77
permutazioni 117
pi greco 34
potere di acquisto 27
probabilità nel gioco del lotto 131
radici reali di una equazione 95
rette 71
risultante di forze concorrenti 156
Ruffini (regola di) 97
scarto quadratico medio 132
scomposizione in fattori 11
sconto 26
Simpson (metodo di) 111
sistemi lineari 102
soluzione delle equazioni algebriche 95
soluzione di un sistema lineare 104
somma di due caratteri 62
somma vettoriale 148
strategie al computer 200
tangente ad una curva in un punto 88
tartaglia (triangolo di) 143
teorema dei seni 49
teorema di Pitagora 43
terne pitagoriche 43
tiri balistici 169
triangolazione 49
unità di misura 181



Siete interessati in personal computer?

*Su questo argomento, nella collana "il piacere del computer"
sono stati pubblicati tra gli altri i seguenti titoli.*

32 programmi con il PET

Il volume contiene 32 programmi in Basic, completamente documentati con listati, esecuzioni di prova, istruzioni per far girare il programma, suggerimenti per variazioni, ecc. Tutti i programmi sono stati verificati e possono essere eseguiti su ogni tipo di PET.

240 pagine, 12.000 lire, sigla PDC 1

Intervista sul personal computer, hardware

Seicento domande e risposte sul mondo del personal computer. Questo primo volume, dedicato all'hardware, contiene una introduzione, sotto forma di intervista, ai computer in generale e ai microprocessori in particolare.

240 pagine, 12.000 lire, sigla PDC 2

32 programmi con l'Apple II

Il volume contiene 32 programmi in Basic, completamente documentati con listati, esecuzioni di prova, istruzioni per far girare il programma, suggerimenti per variazioni, ecc. Tutti i programmi sono stati verificati e possono essere eseguiti su ogni tipo di Apple II.

248 pagine, 12.000 lire, sigla PDC 3

Microsoft Basic

Un breve manuale di introduzione al Microsoft Basic, il Basic dei personal computer. Oggi il Microsoft Basic, una evoluzione e specializzazione del Basic originale, è di fatto lo standard del Basic per microcomputer.

150 pagine, 12.000 lire, sigla PDC 4

Pascal

Questo testo è scritto per coloro che non hanno esperienza di calcolatori o programmazione. Gli argomenti sono organizzati in modo che il lettore possa iniziare a programmare fin dall'inizio.

200 pagine, 12.000 lire, sigla PDC 5

32 programmi con il TRS-80

Trentadue programmi, completamente documentati, pronti per essere eseguiti su un TRS-80 modello I. Il volume comprende i listati, le spiegazioni e i consigli per ulteriori progetti.

240 pagine, 12.000 lire, sigla PDC 6

Intervista sul personal computer, software

In questo secondo volume, dedicato al software, altre centinaia di domande e risposte sul mondo del personal computer. Contiene una introduzione alla programmazione, ai linguaggi assembler e a quelli evoluti.

200 pagine, 12.000 lire, sigla PDC 7

Imparate il Basic con il PET/IBM

Questo libro è stato progettato per essere utile a chiunque desideri imparare a programmare in Basic avendo a disposizione un PET. Studenti di ogni livello, professionisti e dilettanti possono trarre profitto da questo libro e prendere contatto immediato con il mondo dei personal computer.

250 pagine, 12.000 lire, sigla PDC 8

*Troverete questi libri nelle principali librerie,
oppure potete ordinarli direttamente alla casa editrice
compilando la cartolina qui allegata*

**SUA RETRO
AL THE
INFORMAZIONI**

Ho trovato questa cartolina nel libro

acquistato in

In questo spazio potete scrivere quello che pensate di questo libro.

- ☐ Desidero ricevere il vostro più recente catalogo
☐ Desidero ricevere i volumi qui indicati:

sigla	titolo	prezzo
-------	--------	--------

_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

totale

Il personal computer come professione

Il personal computer vi offre mille opportunità di lavoro: potete scrivere articoli o libri su computer; intraprendere un'attività di consulenza o organizzare un'esposizione locale. Per ognuna di queste attività e per molte altre il libro contiene numerosi consigli per la migliore riuscita.

112 pagine, 9.000 lire, sigla PDC 9

Te ne intendi di computer?

Lo scopo di questo libro è di aumentare il livello della vostra comprensione del computer. Sapere cosa possono e cosa non possono fare. Sapere qual è il loro ruolo nella società, sapere quali problemi creano. Dopo questa lettura, ve ne intenderete di computer.

144 pagine, 8.500 lire, sigla PDC 10

Il Basic e il personal computer, uno: introduzione

Il libro, scritto in tono amichevole e informale, non richiede precedenti esperienze con i computer. Vi è compresa una presentazione del Basic con decine di esempi dettagliati, che fanno diventare realtà le vostre idee.

190 pagine, 14.000 lire, sigla PDC 11

Imparate il linguaggio dell'Apple

Se conoscete già il Basic e possedete un computer Apple, siete sulla strada giusta per diventare un esperto in materia. Questo libro vi mette in grado di comprendere il linguaggio macchina dell'Apple partendo dagli esempi più semplici ed arrivando all'uso del miniassembler.

340 pagine, 15.000 lire, sigla PDC 12

Il Basic e il personal computer, due: applicazioni

Questo volume contiene numerosi programmi in Basic adatti ad ogni personal computer. L'uso delle variabili alfanumeriche, gli algoritmi di sort, i giochi, l'arte con il computer sono esempi di programmi illustrati nel volume.

200 pagine, 14.000 lire, sigla PDC 13

Il manuale del CP/M

Il CP/M è il sistema operativo maggiormente usato per i personal computer. Questo volume è un'introduzione breve ma efficace alla tecnica e alla filosofia del CP/M.

120 pagine, 9.500 lire, sigla PDC 14

A scuola con il PET/IBM

Trenta programmi didattici per ogni tipo di PET/IBM matematica, fisica, statistica, ecc.

160 pagine, 13.000 lire, sigla PDC 15

Il manuale dell'Atari

L'edizione italiana del manuale originale di questo personal computer, ora importato in Italia.

300 pagine, 18.500 lire, sigla PDC 16

Il libro del Commodore VIC 20

Questo libro è per i possessori, i futuri possessori e gli utenti di un personal computer VIC Commodore. È inteso come supplemento al manuale della macchina, e presenta numerose caratteristiche del VIC.

166 pagine, 12.000 lire, sigla PDC 17

Il debug nel personal computer

Il "debug", in informatica, è la messa a punto dei programmi, la ricerca e la correzione finale degli errori. Si tratta di una operazione di grande importanza, che si può effettuare in vari modi, tutti trattati in questo libro.

144 pagine, 15.000 lire, sigla PDC 18

Programmazione in Basic per l'uomo d'affari

Questo libro è per gli uomini d'affari che vogliono imparare ad usare il calcolatore. Il libro comincia con la convinzione che sia più facile per un uomo d'affari imparare a programmare, che per un programmatore imparare l'organizzazione e la gestione di una azienda.

232 pagine, 19.000 lire, sigla PDC 19

Il Sinclair ZX-81

È il microcomputer più venduto nel mondo. Ma come ogni computer, necessita di software e documentazione: questo libro contiene 37 programmi che illustrano tutte le caratteristiche e le capacità della macchina.

160 pagine, 13.000 lire, sigla PDC 20

Dal Basic al Pascal

Questa eccezionale guida rende facile per chiunque passare dalla propria conoscenza del Basic ad una perfetta padronanza del Pascal, un linguaggio a 2 livelli che consente di accelerare il tempo reale di operazione e di controllo della programmazione e che richiede uno spazio inferiore di memoria nel vostro computer.

320 pagine, 19.000 lire, sigla PDC 21

cedola di commissione libraria



franco muzzolo editore

via bonporti, 36

35141 padova

nome

(cognome o o.c. o c.a. o c.a.s.)

indirizzo

c.a.p. o c.a.s.

cognome e nome

il formato della cedola è conforme alle seguenti norme nazionali



franco muzio
editore

via bonporti, 36

35141 padova

nome

cognome

cap città

indirizzo

.....

.....

TRE CASSETTE CON TUTTI I PROGRAMMI

Risparmiate tempo e denaro!

Tutti i programmi descritti in questo libro
sono disponibili già registrati su tre cassette,
pronti per l'uso.

Le tre cassette sono in vendita a
L. 40.000.

Desidero ricevere in contrassegno

☐ le tre cassette
con i programmi di

"A scuola con il Texas TI 99/4A".

Scrivo il mio indirizzo sul retro.

firma

*Scritto da un insegnante, questo è un libro di software per la scuola.
Vi sono contenuti trenta programmi scritti in Basic
per il Texas TI 99/4A home computer.*

*Gli argomenti sono presentati in ordine di difficoltà crescente e sono
tutti generalmente trattati negli studi medi inferiori e superiori.
Ma non mancano applicazioni varie e giochi.*

*Il volume sarà utile agli studenti e agli insegnanti che sentono
l'esigenza di non essere soltanto utilizzatori passivi di questo nuovo
strumento didattico.*

*Risparmiate tempo ed errori!
Tutti i programmi descritti in questo libro sono disponibili
su tre cassette, pronti per essere utilizzati.
Per l'ordine, usate la cartolina inserita nel volume.*



franco muzzio & c. editore

ISBN 88-7021-242-4

L. 18.000 (17.647)

23

Sergio Bonanni

A scuola con il Tema 1900/1